# CANONICAL CORRELATION ANALYSIS IN CROSS-DOMAIN RECOMMENDATION

by

**Shaghayegh Sahebi**

B.Sc. in Computer Engineering, Sharif University of Technology, 2005

M.Sc. in Computer Engineering, University of Tehran, 2009

M.Sc. in Intelligent Systems Program, University of Pittsburgh, 2013

Submitted to the Graduate Faculty of
the Kenneth P. Dietrich School of Arts and Sciences
in partial fulfillment of the requirements for the degree of

**Doctor of Philosophy**

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

KENNETH P. DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Shaghayegh Sahebi

It was defended on

July 21st 2016

and approved by

Peter Brusilovsky, School of Information Sciences

Gregory Cooper, Department of Biomedical Informatics

Yu-Ru Lin, School of Information Sciences

Alexander Tuzhilin, Stern School of Business, NYU

Dissertation Director: Peter Brusilovsky, School of Information Sciences

# CANONICAL CORRELATION ANALYSIS IN CROSS-DOMAIN RECOMMENDATION

Shaghayegh Sahebi, PhD

University of Pittsburgh, 2016

Cross-domain recommendation has recently emerged as a hot topic in the field of recommender systems. The idea is to use rating information accumulated in one domain (known as a source or auxiliary domain) to improve the quality of recommendations in another domain (known as a target domain). One of the important problems in cross-domain recommendation is the selection of source domains appropriate for a target domain. Previous works mostly assume that the best domain pairs can be decided based on similarity of their nature (such as books and movies), or simulate domain pairs by splitting the same dataset into multiple domains. We argue that the success of cross-domain recommendations depends on domain characteristics and shared (latent) information among domains; therefore posing new questions: What makes a good auxiliary domain? How should we choose the best auxiliary domain for a specific target domain? In this dissertation we examine the success and failure of cross-domain collaborative filtering across three different datasets with various characteristics of domains. Our goals are to explore the added value of cross-domain recommendations in comparison with traditional within-domain recommendations, and to achieve some progress in uncovering the main mystery of cross-domain recommendation: how can we determine whether a pair of domains is a good candidate for applying cross-domain recommendation techniques? For the former goal, we propose a cross-domain collaborative filtering approach based on canonical correlation analysis. In order to address the latter goal, we investigate a canonical correlation approach as a possible predictor of successful domain pairs and examine a range of features of a single domain and domain pairs in order to see how they could be used to improve predictions. Eventually, we propose a domain-pair classifier that can distinguish between the beneficial and non-beneficial domain pairs before performing the recommendations.

## TABLE OF CONTENTS

PREFACE . . . . . . . . xxiv

1.0 INTRODUCTION . . . . . . . . 1

- 1.1 Challenges and Motivations . . . . . . . . 2
- 1.2 Objectives and Problem Statement . . . . . . . . 4
- 1.3 Organization of the Thesis . . . . . . . . 8

2.0 BACKGROUND AND RELATED WORK . . . . . . . . 9

- 2.1 Notation used in this document . . . . . . . . 9
- 2.2 Single and Cross-Domain Recommender Systems . . . . . . . . 10
  - 2.2.1 Content-Based Cross-Domain Recommendations . . . . . . . . 12
  - 2.2.2 Collaborative Filtering Cross-Domain Recommenders . . . . . . . . 13
  - 2.2.3 Hybrid Cross-Domain Recommendations . . . . . . . . 16
  - 2.2.4 Shared Data in Cross-Domain Recommenders . . . . . . . . 18
  - 2.2.5 Selecting the Best Auxiliary Domain . . . . . . . . 18
  - 2.2.6 Related Collaborative Filtering Approaches . . . . . . . . 19
- 2.3 Canonical Correlation Analysis . . . . . . . . 20
  - 2.3.1 Regularized CCA . . . . . . . . 20
  - 2.3.2 Large-Scale CCA . . . . . . . . 22
  - 2.3.3 CCA in Recommender Systems . . . . . . . . 23

3.0 CCA-BASED CROSS-DOMAIN ALGORITHMS . . . . . . . . 25

- 3.1 CD-CCA . . . . . . . . 25
- 3.2 CD-LCCA . . . . . . . . 28
- 3.3 Baseline Algorithms . . . . . . . . 32

3.3.1 SVD++ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
3.3.1.1 SD-SVD and CD-SVD . . . . . . . . . . . . . . . . . . . 33
3.3.2 Rating Matrix Generative Model (RMGM) . . . . . . . . . . . . . . 33
3.3.3 Collective Matrix Factorization (CMF) . . . . . . . . . . . . . . . . 34
3.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
**4.0 DATASETS** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
4.1 Yelp Academic Dataset . . . . . . . . . . . . . . . . . . . . . . . . 38
4.2 Imhonet Dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.3 Supermarket Dataset . . . . . . . . . . . . . . . . . . . . . . . . . 42
4.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
**5.0 GENERAL EXPERIMENTS: CD-CCA VS. BASELINE ALGORITHMS** 49
5.1 Experiment Setup . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
5.2 Results of the Supermarket Purchase Dataset . . . . . . . . . . . . . . . . 53
5.3 Results of the Yelp Dataset . . . . . . . . . . . . . . . . . . . . . . 59
5.4 Results of the Imhonet Dataset . . . . . . . . . . . . . . . . . . . . 69
5.5 Performance Analysis of Algorithms . . . . . . . . . . . . . . . . . . 73
5.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
**6.0 COLD-START EXPERIMENTS** . . . . . . . . . . . . . . . . . . . . 80
6.1 Experiment Setup . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
6.2 Cold-Start Analysis for the Supermarket Purchase Dataset . . . . . . . . . 81
6.3 Cold-Start Analysis for the Yelp Dataset . . . . . . . . . . . . . . . . 88
6.4 Cold-Start Analysis for the Imhonet Dataset . . . . . . . . . . . . . . . 92
6.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 99
**7.0 FINDING THE APPROPRIATE AUXILIARY DOMAIN** . . . . . . . . 102
7.1 Data Characteristics . . . . . . . . . . . . . . . . . . . . . . . . . 103
7.2 Correlation Analysis . . . . . . . . . . . . . . . . . . . . . . . . . 104
7.2.1 Correlation Analysis for the Supermarket Purchase Dataset . . . . 105
7.2.1.1 Correlation Analysis of Errors . . . . . . . . . . . . . . . . 105
7.2.1.2 Correlation Analysis of Improvement Ratio . . . . . . . . . . 109
7.2.2 Correlation Analysis for Yelp Dataset . . . . . . . . . . . . . . . . . 112

7.2.2.1 Correlation Analysis of Errors . . . . . . . . . . . . . . . . . 112

7.2.2.2 Correlation Analysis of Improvement Ratio . . . . . . . . . 116

7.2.3 Correlation Analysis for Imhonet Dataset . . . . . . . . . . . . . . . 119

7.2.3.1 Correlation Analysis of Errors . . . . . . . . . . . . . . . . . 119

7.2.3.2 Correlation Analysis of Improvement Ratio . . . . . . . . . 122

7.2.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 125

7.3 Regression Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 126

7.3.1 Experiment Setup . . . . . . . . . . . . . . . . . . . . . . . . . 127

7.3.2 Regression Analysis for the Supermarket Dataset . . . . . . . . . . 127

7.3.3 Regression Analysis for the Yelp Dataset . . . . . . . . . . . . . . . 131

7.3.4 Regression Analysis for the Imhonet Dataset . . . . . . . . . . . . . 132

7.3.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 133

7.4 Nature of Good Domain-Pairs . . . . . . . . . . . . . . . . . . . . . . . 135

7.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 139

**8.0 AUXILIARY DOMAIN CLASSIFIER** . . . . . . . . . . . . . . . . . . . 142

8.1 Experiment Setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142

8.2 Within Dataset Classification . . . . . . . . . . . . . . . . . . . . . . . 145

8.3 Cross Dataset Classification . . . . . . . . . . . . . . . . . . . . . . . 148

8.4 Mixed Dataset Classification . . . . . . . . . . . . . . . . . . . . . . . 150

8.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153

**9.0 CONCLUSIONS AND DISCUSSION** . . . . . . . . . . . . . . . . . . . 155

9.1 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155

9.2 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159

9.3 Limitations and Delimitations . . . . . . . . . . . . . . . . . . . . . . . 160

9.4 Future Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 163

**APPENDIX A. SUPERMARKET DATA FIGURES AND TABLES** . . . . 165

A.1 Supermarket Data Domain Mapping . . . . . . . . . . . . . . . . . . . 165

A.2 Domain Pair Statistics for the Supermarket Dataset . . . . . . . . . . . . 171

A.3 Error of Algorithms on Domain Pairs in Supermarket Dataset . . . . . . 177

A.4 Cold-Start Analysis of Domain Pairs for Supermarket Dataset . . . . . . 183

A.4.1 MAEs for Target User Profiles . . . . . . . . . . . . . . . . . . . . 183

A.4.2 RMSEs for Target User Profiles . . . . . . . . . . . . . . . . . . . 186

A.4.3 RMSEs for Source User Profiles . . . . . . . . . . . . . . . . . . . 189

A.4.4 RMSEs for Source User Profiles . . . . . . . . . . . . . . . . . . . 192

A.5 Correlation Analysis for Supermarket Dataset . . . . . . . . . . . . . . . . 195

A.5.1 Correlation Analysis Plots for the RMSE of Algorithms in the Supermarket Dataset . . . . . . . . . . . . . . . . . . . . . . . . . 195

A.5.2 Correlation Analysis Plots for the Improvement Ratio of Algorithms in the Supermarket Dataset . . . . . . . . . . . . . . . . . . . . 196

**APPENDIX B. YELP DATA FIGURES AND TABLES** . . . . . . . . . . . 204

B.1 Domain Pair Statistics for Yelp Dataset . . . . . . . . . . . . . . . . . . . 204

B.2 Error of Algorithms on Domain Pairs in Yelp Dataset . . . . . . . . . . . . 229

B.3 Cold-Start Analysis of Domain Pairs for Yelp Dataset . . . . . . . . . . . 235

B.3.1 MAEs for Target User Profiles . . . . . . . . . . . . . . . . . . . . 235

B.3.2 MAEs for Source User Profiles . . . . . . . . . . . . . . . . . . . . 244

B.4 Correlation Analysis for Yelp Dataset . . . . . . . . . . . . . . . . . . . . 253

B.4.1 Correlation Analysis Plots for the RMSE of Algorithms in the Yelp Dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 253

B.4.2 Correlation Analysis Plots for the Improvement Ratio of Algorithms in the Yelp Dataset . . . . . . . . . . . . . . . . . . . . . . . . 256

**APPENDIX C. IMHONET DATA FIGURES AND TABLES** . . . . . . . . 264

C.1 Domain Pair Statistics for Imhonet Dataset . . . . . . . . . . . . . . . . . 264

C.2 Error of Algorithms on Domain Pairs in Imhonet Dataset . . . . . . . . . . 267

C.3 Cold-Start Analysis of Domain Pairs for Imhonet Dataset . . . . . . . . . . 268

C.3.1 MAEs for Target User Profiles . . . . . . . . . . . . . . . . . . . . 268

C.3.2 RMSEs for Target User Profiles . . . . . . . . . . . . . . . . . . . 270

C.3.3 MAEs for Source User Profiles . . . . . . . . . . . . . . . . . . . . 272

C.3.4 RMSEs for Source User Profiles . . . . . . . . . . . . . . . . . . . 274

C.4 Correlation Analysis for Imhonet Dataset . . . . . . . . . . . . . . . . . . 276

C.4.1 Correlation Analysis Plots for the RMSE of Algorithms in the Imhonet Dataset . . . 276
C.4.2 Correlation Analysis Plots for the Improvement Ratio of Algorithms in the Imhonet Dataset . . . 279
**BIBLIOGRAPHY** . . . 284

## LIST OF TABLES

1 Proposed and baseline algorithms and their properties . . . . . . . . . . . . . 37

2 General Characteristics of Datasets . . . . . . . . . . . . . . . . . . . . . . . 39

3 Basic Statistics for Yelp Academic Dataset. . . . . . . . . . . . . . . . . . . . 40

4 Basic Statistics for Imhonet Dataset. . . . . . . . . . . . . . . . . . . . . . . . 41

5 Basic Statistics for Supermarket Dataset . . . . . . . . . . . . . . . . . . . . . 44

6 Number of domain pairs with significant RMSE difference among algorithms. Each row shows the number of domain pairs in which the algorithm of that row works significantly better than the algorithms mentioned in the columns. 57

7 Number of domain pairs with significant MAE difference among algorithms. Each row shows the number of domain pairs in which the algorithm of that row works significantly better than the algorithms mentioned in the columns. 57

8 Correlation between RMSE of algorithms on all domain pairs in the Supermarket dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60

9 Summary of domain pair statistics in Yelp dataset . . . . . . . . . . . . . . . 61

10 Number of domain pairs with significant RMSE difference among algorithms for the Yelp dataset. Each row shows the number of domain pairs in which the algorithm of that row works significantly better than the algorithms mentioned in the columns. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65

11 Number of domain pairs with significant MAE difference among algorithms for the Yelp dataset. Each row shows the number of domain pairs in which the algorithm of that row works significantly better than the algorithms mentioned in the columns. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67

12 Correlation of RMSE of algorithms with each other. *: significant with p-value $<$ 0.01; **: significant with p-value $<$ 0.001; ***: significant with p-value $<$ 0.0001 67

13 Number of domain pairs with significant error difference among algorithms for the Imhonet dataset. Each row shows the number of domain pairs in which the algorithm of that row works significantly better than the algorithms mentioned in the columns. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71

14 Correlation of RMSE of algorithms with each other in the Imhonet dataset. *: significant with p-value $<$ 0.01. . . . . . . . . . . . . . . . . . . . . . . . . . . 73

15 Average running time of each algorithm on one domain pair in the Supermarket dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76

16 Correlations of data characteristics with MAE of algorithms on the Supermaket dataset; *: significant with p-value $<$ 0.05; **: significant with p-value $<$ 0.01; ***: significant with p-value $<$ 0.001 . . . . . . . . . . . . . . . . . . . . . . 106

17 Correlations of data characteristics with RMSE of algorithms on the Supermaket dataset; *: significant with p-value $<$ 0.05; **: significant with p-value $<$ 0.01; ***: significant with p-value $<$ 0.001 . . . . . . . . . . . . . . . . . 107

18 Correlations of data characteristics with MAE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Supermarket dataset; *: significant with p-value $<$ 0.05; **: significant with p-value $<$ 0.01; ***: significant with p-value $<$ 0.001 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110

19 Correlations of data characteristics with RMSE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Supermarket dataset; *: significant with p-value $<$ 0.05; **: significant with p-value $<$ 0.01; ***: significant with p-value $<$ 0.001 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111

20 Correlations of data characteristics with MAE of algorithms on the Yelp dataset; *: significant with p-value $<$ 0.05; **: significant with p-value $<$ 0.01; ***: significant with p-value $<$ 0.001 . . . . . . . . . . . . . . . . . . . . . . . . . 113

21 Correlations of data characteristics with RMSE of algorithms on the Yelp dataset; *: significant with p-value $<$ 0.05; **: significant with p-value $<$ 0.01; ***: significant with p-value $<$ 0.001 . . . . . . . . . . . . . . . . . . . . . . 114

22 Correlations of data characteristics with MAE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Yelp dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117

23 Correlations of data characteristics with RMSE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Yelp dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118

24 Correlations of data characteristics with MAE of algorithms on the Imhonet dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001 . . . . . . . . . . . . . . . . . . . . . . . 120

25 Correlations of data characteristics with RMSE of algorithms on the Imhonet dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001 . . . . . . . . . . . . . . . . . . . . . . . 121

26 Correlations of data characteristics with MAE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Imhonet dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123

27 Correlations of data characteristics with RMSE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Imhonet dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124

28 RMSE regression analysis results for the Supermaket dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 128

29 Improvement ratio regression analysis results for the Supermaket dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001 . . . . . . . . . . . . . . . . . . . . . . . . 130

30 RMSE regression analysis results for the Yelp dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$ . . . . . 131

31 Improvement ratio regression analysis results for the Yelp dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$ . . . . . 132

32 RMSE regression analysis results for the Imhonet dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$ . . . . . 133

33 Improvement ratio regression analysis results for the Imhonet dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$ . . . . . 134

34 Percentage of positive class labels in each dataset . . . . . 145

35 Percentage of positive class labels in each mixed dataset . . . . . 151

36 Mapping of supermarket item categories to domains . . . . . 165

37 Domain and domain-pair data size statistics for the supermarket dataset . . . 171

38 Each domain ratings' central tendency and dispersion statistics for the supermarket dataset . . . . . 173

39 Each domain and domain-pair ratings' dispersion statistics for the supermarket dataset . . . . . 175

40 RMSE and MAE for domain-pairs in the supermarket dataset . . . . . 177

41 Significant RMSE relations of algorithms in each domain pair for the Supermarket dataset . . . . . 179

42 Significant MAE relations of algorithms in each domain pair for the Supermarket dataset . . . . . 181

43 Domain and domain-pair data size statistics for the Yelp dataset . . . . . 204

44 Domain ratings central tendency and dispersion statistics for the Yelp dataset 211

45 Domain and domain-pair ratings dispersion statistics for the Yelp dataset . . 217

46 Domain-pair CCA statistics for the Yelp dataset . . . . . 223

47 RMSE and MAE for domain-pairs in the Yelp dataset . . . . . 229

48 Domain and domain-pair data size statistics for the Imhonet dataset . . . . . 264

49 Domain ratings central tendency and dispersion statistics for the Imhonet dataset 265

50 Domain and domain-pair ratings dispersion statistics for the Imhonet dataset 265

51 Domain-pair CCA statistics for the Imhonet dataset . . . . . . . . . . . . . . . 266

52 RMSE and MAE for domain-pairs in the Imhonet dataset . . . . . . . . . . . 267

## LIST OF FIGURES

1 The proposed framework and tasks . . . . . . . . . . . . . . . . . . . . . . . . 7

2 L-CCA algorithm as presented in [44] . . . . . . . . . . . . . . . . . . . . . . 22

3 A toy example of CCA in cross-domain recommender system setting . . . . . 26

4 Mapping between source and target domains in CD-CCA . . . . . . . . . . . 29

5 Cross-Domain Setup for SVD++ . . . . . . . . . . . . . . . . . . . . . . . . . 34

6 Distribution of ratings in the book domain . . . . . . . . . . . . . . . . . . . 43

7 Density of domains in each of the datasets . . . . . . . . . . . . . . . . . . . 46

8 Number of users to number of items ratio in each of the datasets . . . . . . . 47

9 Number of source domain items to number of target domain items in each of the datasets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

10 Experiment setup to answer research question Q1.1 . . . . . . . . . . . . . . . 50

11 Separating test, train, and evaluation data from the target domain . . . . . . 51

12 RMSE of algorithms on 50 Supermarket domain pairs ordered by the RMSE of the CD-CCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55

13 MAE of algorithms on 50 Supermarket domain pairs ordered by the MAE of the CD-CCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56

14 Scatter plot of RMSE of algorithms on 50 Supermarket domain pairs . . . . . 60

15 Scatter plot of MAE of algorithms on 50 Supermarket domain pairs . . . . . 61

16 RMSE of algorithms on 158 Yelp domain pairs ordered by the RMSE of the CD-CCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63

17 MAE of algorithms on 158 Yelp domain pairs ordered by the MAE of the CD-CCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64

18 Scatter plot of RMSE of algorithms on 158 Yelp domain pairs . . . . . . . . . 68

19 Scatter plot of MAE of algorithms on 158 Yelp domain pairs . . . . . . . . . 68

20 RMSE of algorithms on 12 Imhonet domain pairs ordered by the RMSE of the CD-CCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70

21 MAE of algorithms on 12 Imhonet domain pairs ordered by the MAE of the CD-CCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71

22 Scatter plot of RMSE of algorithms on 12 Imhonet domain pairs . . . . . . . 72

23 Scatter plot of MAE of algorithms on 12 Imhonet domain pairs . . . . . . . . 72

24 CD-CCA running time in four sample domain-pairs of the Yelp dataset. Numbers are in logarithmic scale. . . . . . . . . . . . . . . . . . . . . . . . . . 75

25 Target profile sizes of users in the Supermarket dataset . . . . . . . . . . . . 82

26 User-based RMSE of algorithms in the Supermarket dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size . . . . 84

27 User-based MAE of algorithms in the Supermarket dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size . . . . 85

28 Source profile sizes of users in the Supermarket dataset . . . . . . . . . . . . 86

29 User-based RMSE of algorithms in the Supermarket dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size . . . . 87

30 User-based MAE of algorithms in the Supermarket dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size . . . . 88

31 Target profile sizes of users in Yelp dataset . . . . . . . . . . . . . . . . . . . 89

32 User-based RMSE of algorithms in the Yelp dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size . . . . . . . . . 90

33 User-based MAE of algorithms in the Yelp dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size . . . . . . . . . 90

34 Source profile sizes of users in Yelp dataset . . . . . . . . . . . . . . . . . . . 92

35 User-based RMSE of algorithms in the Yelp dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size . . . . . . . . 93

36 User-based MAE of algorithms in the Yelp dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size . . . . . . . . 93

37 Target profile sizes of users in Imhonet dataset . . . . . . . . . . . . . . . . . 94

38 User-based RMSE of algorithms in the Imhonet dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size . . . . . . . . . 95

39 User-based MAE of algorithms in the Imhonet dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size . . . . . . . . . 95

40 Source profile sizes of users in Imhonet dataset . . . . . . . . . . . . . . . . . 97

41 User-based RMSE of algorithms in the Imhonet dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size . . . . . . . . 97

42 User-based MAE of algorithms in the Imhonet dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size . . . . . . . . 98

43 Classification loss in the Supermarket dataset . . . . . . . . . . . . . . . . . . 146

44 Classification loss in the Yelp dataset . . . . . . . . . . . . . . . . . . . . . 147

45 Classification loss in the Imhonet dataset . . . . . . . . . . . . . . . . . . . . 147

46 Classification loss for cross-dataset experiments; trained on the Supermarket data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 149

47 Classification loss for cross-dataset experiments; trained on Yelp data . . . . . 150

48 Classification loss for cross-dataset experiments; trained on Imhonet data . . 151

49 Classification loss in the mix of Supermarket and Yelp datasets . . . . . . . . 152

50 Classification loss in the mix of Supermarket and Imhonet dataset . . . . . . 152

51 Classification loss in the mix of Yelp and Imhonet dataset . . . . . . . . . . . 153

52 User-based MAE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' target domain profile size . . . . . 183

53 User-based RMSE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Target domain profile size . . . . 186

54 User-based MAE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size . . . . 189

55 User-based RMSE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size . . . . 192

56 Scatter plot of general and central tendency statistics with the RMSE of algorithms in the Supermarket dataset . . . . . . . . . . . . . . . . . . . . . . . 195

57 Scatter plot of dispersion statistics with the RMSE of algorithms in the Supermarket dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195

58 Scatter plot of CCA-related statistics with the RMSE of algorithms in the Supermarket dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . 196

59 Scatter plot of general and central tendency statistics with the improvement ratio of CD-CCA over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-CCA is significantly better than othor algorithms 196

60 Scatter plot of dispersion statistics with the improvement ratio of CD-CCA over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-CCA is significantly better than othor algorithms . . . . . . . . . . 197

61 Scatter plot of CCA-related statistics with the improvement ratio of CD-CCA over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-CCA is significantly better than othor algorithms . . . . . . . . . . 197

62 Scatter plot of general and central tendency statistics with the improvement ratio of CD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-SVD is significantly better than othor algorithms 198

63 Scatter plot of dispersion statistics with the improvement ratio of CD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-SVD is significantly better than othor algorithms . . . . . . . . . . 198

64 Scatter plot of CCA-related statistics with the improvement ratio of CD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-SVD is significantly better than othor algorithms . . . . . . . . . . 199

65 Scatter plot of general and central tendency statistics with the improvement ratio of SD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which SD-SVD is significantly better than othor algorithms 199

66 Scatter plot of dispersion statistics with the improvement ratio of SD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which SD-SVD is significantly better than othor algorithms . . . . . . . . . . 200

67 Scatter plot of CCA-related statistics with the improvement ratio of SD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which SD-SVD is significantly better than othor algorithms . . . . . . . . . . 200

68 Scatter plot of general and central tendency statistics with the improvement ratio of CMF over other algorithms in the Supermarket dataset; red cross shows the cases in which CMF is significantly better than othor algorithms . 201

69 Scatter plot of dispersion statistics with the improvement ratio of CMF over other algorithms in the Supermarket dataset; red cross shows the cases in which CMF is significantly better than othor algorithms . . . . . . . . . . . . . . . 201

70 Scatter plot of CCA-related statistics with the improvement ratio of CMF over other algorithms in the Supermarket dataset; red cross shows the cases in which CMF is significantly better than othor algorithms . . . . . . . . . . . 202

71 Scatter plot of general and central tendency statistics with the improvement ratio of RMGM over other algorithms in the Supermarket dataset; red cross shows the cases in which RMGM is significantly better than othor algorithms 202

72 Scatter plot of dispersion statistics with the improvement ratio of RMGM over other algorithms in the Supermarket dataset; red cross shows the cases in which RMGM is significantly better than othor algorithms . . . . . . . . . . . . . . 203

73 Scatter plot of CCA-related statistics with the improvement ratio of RMGM over other algorithms in the Supermarket dataset; red cross shows the cases in which RMGM is significantly better than othor algorithms . . . . . . . . . . . 203

74 User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' target domain profile size . . . . . . . . . 235

75 User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' source domain profile size . . . . . . . . . 244

76 Scatter plot of general and central tendency statistics with the RMSE of algorithms in the Yelp dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . 253

77 Scatter plot of dispersion statistics with the RMSE of algorithms in the Yelp dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 253

78 Scatter plot of CCA-related statistics with the RMSE of algorithms in the Yelp dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 254

79 Scatter plot of general and central tendency statistics with the MAE of algorithms in the Yelp dataset . . . . . . . . . . . . . . . . . . . . . . . . . . 254

80 Scatter plot of dispersion statistics with the MAE of algorithms in the Yelp dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 255

81 Scatter plot of CCA-related statistics with the MAE of algorithms in the Yelp dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 255

82 Scatter plot of general and central tendency statistics with the improvement ratio of CD-CCA over other algorithms in the Yelp dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms . . . 256

83 Scatter plot of dispersion statistics with the improvement ratio of CD-CCA over other algorithms in the Yelp dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms . . . . . . . . . . . . 256

84 Scatter plot of CCA-related statistics with the improvement ratio of CD-CCA over other algorithms in the Yelp dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms . . . . . . . . . . . . 257

85 Scatter plot of general and central tendency statistics with the improvement ratio of CD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms . . . 257

86 Scatter plot of dispersion statistics with the improvement ratio of CD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms . . . . . . . . . . . . . 258

87 Scatter plot of CCA-related statistics with the improvement ratio of CD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms . . . . . . . . . . . . . 258

88 Scatter plot of general and central tendency statistics with the improvement ratio of SD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms . . . . . 259

89 Scatter plot of dispersion statistics with the improvement ratio of SD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms . . . . . . . . . . . . . . 259

90 Scatter plot of CCA-related statistics with the improvement ratio of SD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms . . . . . . . . . . . . . . 260

91 Scatter plot of general and central tendency statistics with the improvement ratio of CMF over other algorithms in the Yelp dataset; red cross shows the cases in which CMF is significantly better than other algorithms . . . . . . . 260

92 Scatter plot of dispersion statistics with the improvement ratio of CMF over other algorithms in the Yelp dataset; red cross shows the cases in which CMF is significantly better than other algorithms . . . . . . . . . . . . . . . . . . . 261

93 Scatter plot of CCA-related statistics with the improvement ratio of CMF over other algorithms in the Yelp dataset; red cross shows the cases in which CMF is significantly better than other algorithms . . . . . . . . . . . . . . . . . . . 261

94 Scatter plot of general and central tendency statistics with the improvement ratio of RMGM over other algorithms in the Yelp dataset; red cross shows the cases in which RMGM is significantly better than other algorithms . . . . . . 262

95 Scatter plot of dispersion statistics with the improvement ratio of RMGM over other algorithms in the Yelp dataset; red cross shows the cases in which RMGM is significantly better than other algorithms . . . . . . . . . . . . . . . . . . . 262

96 Scatter plot of CCA-related statistics with the improvement ratio of RMGM over other algorithms in the Yelp dataset; red cross shows the cases in which RMGM is significantly better than other algorithms . . . . . . . . . . . . . . 263

97 User-based MAE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' target domain profile size . . . . . . . . . 268

98 User-based RMSE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' target domain profile size . . . . . . . . . 270

99 User-based MAE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size . . . . . . . . . 272

100 User-based RMSE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size . . . . . . . . 274

101 Scatter plot of general and central tendency statistics with the RMSE of algorithms in the Imhonet dataset . . . . . . . . . . . . . . . . . . . . . . . . 276

102 Scatter plot of dispersion statistics with the RMSE of algorithms in the Imhonet dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 276

103 Scatter plot of CCA-related statistics with the RMSE of algorithms in the Imhonet dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 277

104 Scatter plot of general and central tendency statistics with the MAE of algorithms in the Imhonet dataset . . . . . . . . . . . . . . . . . . . . . . . . 277

105 Scatter plot of dispersion statistics with the MAE of algorithms in the Imhonet dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 278

106 Scatter plot of CCA-related statistics with the MAE of algorithms in the Imhonet dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 278

107 Scatter plot of general and central tendency statistics with the improvement ratio of CD-CCA over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms . . . 279

108 Scatter plot of dispersion statistics with the improvement ratio of CD-CCA over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms . . . . . . . . . . 279

109 Scatter plot of CCA-related statistics with the improvement ratio of CD-CCA over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms . . . . . . . . . . 280

110 Scatter plot of general and central tendency statistics with the improvement ratio of CD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms . . . 280

111 Scatter plot of dispersion statistics with the improvement ratio of CD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms . . . . . . . . . . 281

112 Scatter plot of CCA-related statistics with the improvement ratio of CD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms . . . . . . . . . . 281

113 Scatter plot of general and central tendency statistics with the improvement ratio of SD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms . . . 282

114 Scatter plot of dispersion statistics with the improvement ratio of SD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms . . . . . . . . . . . . . . 282

115 Scatter plot of CCA-related statistics with the improvement ratio of SD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms . . . . . . . . . . 283

# PREFACE

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Peter Brusilovsky, director of the Personalized Adaptive Web Systems (PAWS) lab, for his constant advice, encouragement, and academic and emotional support throughout my PhD studies. Thank you for guiding me while still giving me freedom to select my thesis topic, for opening doors to fruitful internships and collaborations, and for your patience while I was working remotely on this thesis. Without your time and support, this thesis would not have become a reality.

I would like to express my very sincere gratitude to the others on my thesis committee: Prof. Gregory Cooper, Prof. Yu-Ru Lin, and Prof. Alexander Tuzhilin, for their invaluable comments, questions, and encouragement that added new perspectives to this thesis and my work. Thank you for taking the time to serve on this thesis committee despite your own deadlines and travel schedules.

This thesis would not have been possible without access to the datasets analyzed within it. I would like to thank Dr. Shlomo Berkovsky for providing the Supermarket dataset as well as his valuable insights. I must also acknowledge Imhonet.ru for entrusting their data to our research team in PAWS.

My sincere thanks also goes to my mentors and professors, Dr. Trevor Walker, Prof. William Cohen, Prof. Rosta Farzan, and Prof. Daqing He, all of whom supported me greatly during my doctoral studies.

I am grateful to my friends at the PAWS lab and the Intelligent Systems Program for all the encouragement, discussions, and moral support during my PhD years. Thank you for always being there for me, listening to my practice talks and providing me with critical feedback. During these years of earning my doctoral degree, PAWS lab lunches with colleagues were the most delightful part of my days.

I would like to thank my beloved parents and sisters for their unconditional love, sacrifice, and continuous support through out my entire life. Thank you for supporting me while I pursue my dreams in a country $11,000$ kilometers away from home, knowing that we will not be able to visit each other for several years.

Last, but certainly not least, I would like to thank the love of my life, my best friend and husband, Amir, for his unfailing love, encouragement, and assistance. Thank you for all you have done for me, especially when times were tough. Thank you for being my pillar of strength, my wise counsel, and my sympathetic ear.

# 1.0 INTRODUCTION

Information overload is one of the byproducts of internet growth. As the amount of information increases on the web, users face the challenge of finding the most relevant and useful information and products on the internet. Information filtering approaches and systems, such as information retrieval approaches, search engines, and faceted search interfaces, aim to alleviate this problem by either finding the most relevant information for users or helping users to find the most relevant information in a fast and efficient way. Recommender systems (or recommendation systems) have emerged as one of the solutions to the information overload problem in 1990's [22].

The goal of recommender systems is to present the most desirable information and products (items) to users based on their preferences. Many of the recommender systems rely on user modeling approaches to approximate user preferences and build user profiles based on users' purchase history, click-stream logs, item ratings, etc. For example, collaborative-filtering recommender systems achieve their goal by finding users of a similar taste, create user models built on users' histories, and recommend the items liked by these users to each other[1].

Although recommender systems are the topic of much research[2] and are very common in commercial systems, they still suffer from many problems. One of the important problems in recommender systems is the cold-start problem. When a new user starts using the system, or a new item is introduced in it, there is little to no history that is known for that user

[1] Of course this is a very simplistic and high-level interpretation of the idea behind collaborative filtering. Each of the collaborative filtering approaches have their own elaborated algorithm to implement such a general idea.

[2] As examples, one can refer to machine learning and user modeling conferences, such as RecSys, UMAP, and KDD.

or item. As a result, the recommender system that relies on user or item histories cannot build a reliable profile for that user or item. Consequently, this recommender system will not be able to recommend any items to the cold-start user or cannot recommend the cold-start item to any users. A whole body of recommender system literature aims to target the cold-start problem. As examples, we can name hybrid recommender systems [8], context-based recommender systems [2], and community-based recommender systems [61]. Cross-domain recommender systems [6] are one of the newest types of systems that promise to alleviate the cold-start problem.

Cross-domain recommendation has recently emerged as a hot topic in the field of recommender systems [20]. Their idea is to use rating information accumulated in one domain (known as a source or auxiliary domain) to improve the quality of recommendations in another domain (known as a target domain). The proponents of cross-domain recommendation claim that such a technique can be especially helpful when a user has few or no ratings in the target domain or when the quality of recommendation in the target domain is low, due to lack of other information. A modern user works with many systems and many information domains. While she may have a solid user profile in a system that she has previously used, beginning to use a new domain or system would potentially benefit from cross-domain information. In this thesis, we aim to examine the promise of cross-domain collaborative filtering recommender systems, by studying their feasibility and performance, understanding when they work well, and finding the best domains in which these recommenders work well. In the following sections, we introduce the challenges of cross-domain collaborative filtering systems, our problem statement and research questions, and the organization of this thesis.

## 1.1 CHALLENGES AND MOTIVATIONS

As mentioned earlier, cross-domain recommender systems promise to alleviate cold-start problem, provide better quality recommendations, and provide a better understanding of user preferences by transferring information from one or multiple source domains to the target domain. However, little of the research on cross-domain recommender systems has studied these issues comprehensively. One of the main reasons for this circumstance is the difficulty of finding real-world cross-domain datasets. Most of the systems that have multiple domains of items are commercial systems. Many of these systems have limitations in providing their users' private data to the researchers.

To avoid this problem, many researchers have simulated different domains in one dataset by splitting the items into multiple sectors, and treating each sector as a domain. This split can be based on some item features, e.g. genre for movies, or totally random [6]. In these cases, the generalizability of approaches are not tested on real domains and their distributions. For example, it is more likely that two different genres of movies have a similar distribution of user ratings, compared to rating distributions of the movie and electronic products domains. Users pay attention to the similar characteristics of movies, such as actors, screenplay, and director, in addition to the genre, when they want to rate them. But the factors for rating an electronic product is completely different from the factors to rate a movie. For example, in electronic products, the year the product is built is a very important factor; most people would like to have the most recent products in that domain. But, there are some old movies that will always be on top of peoples' favorite lists. So, eventually if an approach is tested on different genres of movies, it will not necessarily work for every set of domains.

Another important problem in cross-domain recommendation is the selection of source domains appropriate for a target domain. Previous works mostly assume that the best domain pairs can be decided based on similarity of their nature (such as books and movies) [76, 60], or simulate domain pairs by splitting the same dataset into multiple domains [6]. While the majority of early works have typically focused on one or two pairs of intuitively related domains and return quite positive results, which confirms the hopes of cross-domain enthusiasts [7, 60, 76], there also exist mixed results for cross-domain recommendations [64]. We argue that the success of cross-domain recommendations depends on domain characteristics and shared (latent) information among domains, in addition to the cross-domain algorithm itself; therefore posing new questions: What makes a good auxiliary domain? Do the features used to define a good auxiliary domain work for all algorithms and all datasets? and How should we choose the best auxiliary domain for a specific target domain?

In the next section, we introduce the research questions that we study in this thesis, to work towards resolving these issues in cross-domain collaborative filtering.

## 1.2 OBJECTIVES AND PROBLEM STATEMENT

In this thesis, we aim to gain a more detailed understanding of cross-domain collaborative filtering and the factors that affect its performance. More specifically, we would like to study if cross-domain recommendations are feasible and beneficial and what can lead us to the auxiliary domains that provide the most benefit in cross-domain recommenders.

The improvement we achieve from cross-domain recommendation could result from the added information or the approach we employ to recommend items in the target domain. As discussed in [10] some methods can improve the recommendation result without transferring knowledge from source to target domain, and only by improving the algorithm of recommendation. On the other hand, sometimes just adding the extra information from a related source domain and using a single-domain algorithm on the extended data can also improve the recommendations, especially in the cold-start setting [60]. In this thesis, we propose to study each of these aspects: if the improvement we get from cross-domain recommendations are because of the extra information added by the auxiliary domain; or because of the algorithm used in the cross-domain recommender; or for both of these reasons.

Another important factor in the results of cross-domain recommendation is the selection of the auxiliary domain. As discussed in the previous section, most of the work in the literature are either based on the assumption of choosing naturally similar domains or subdividing a single domain into multiple simulated domains. However, we expect some auxiliary domains to be better choices compared to others. We hypothesize that some data characteristics, in addition to the nature of the auxiliary domain, can be important factors in choosing better source domains.

In addition to finding out if an auxiliary domain helps or not, we hypothesize that the amount of improvement we get from performing cross-domain recommendation using each pair of domains can be affected by some of the data characteristics. We expect that discovering the key characteristics of a good auxiliary-target domain pair, leads us to establish an auxiliary domain classifier that can distinguish between beneficial and non-beneficial source domains for a specific target domain. Eventually, we hypothesize that we can find interesting relationships between the domains for cross-domain recommendation. We would like to know if the discovered domains are also “intuitively” related.

In summary, here are the questions we would like to study in this thesis:

- Q1: Is cross-domain recommendation feasible and beneficial?
  - Q1.1: Is the benefit gained from a cross-domain recommender because of the extra data or the used approach?
  - Q1.2: Does cross-domain recommendation benefit the cold-start situation?
- Q2: Are all source domains helpful to all target domains?
  - Q2.1: What are the factors that distinguish between a helpful source domain and a non-helpful one?
  - Q2.2: What are the factors that determine the amount of improvement we get using cross-domain recommendation based on specific source and target domains?
  - Q2.3: What is the nature of good domain-pair choices?
- Q3: Is classification of domain-pairs into beneficial and non-beneficial feasible?

To answer the above research questions, we propose to use Canonical Correlation Analysis (CCA) as a tool for both performing recommendations and distinguishing the helpful auxiliary domains. To study the feasibility and benefit of cross-domain recommenders (Q1), we propose a cross-domain recommender approach based on canonical correlation analysis (CD-CCA), including a large-scale implementation of it (CD-LCCA). We compare CD-CCA with other cross-domain algorithms as baselines to study the approach effect on cross-domain recommendation results. We run a single-domain algorithm on target domain data to research the effect of added information compared to cross-domain approaches (Q1.1). To answer research question Q1.2, we analyze the results based on the size of user profiles.

In addition to using CCA for delivering cross-domain recommendations, we propose to use it in finding the beneficial auxiliary domains. We define various factors extracted from CCA, such as average CCA correlation in all components and number of significant canonical correlations between components, as cross-domain data factors in our analysis. In addition to that, we use other cross-domain characteristics, such as KL-divergence, and other single-domain features, such as data sparsity, in our study. We experiment on three datasets with different natures and various number of domains to study different combinations of source and target domains. We perform both bivariate correlation analysis and multi-variable regression analysis on these factors to answer the the second research question and its sub-questions. To study Q2.2, we define an "Improvement Ratio" factor as a dependent variable and run bivariate correlation analysis and multi-variable regression on data characteristics. We study the qualifications of domain-pairs and their characteristics in more details to find out about the counter-intuitive domain-pairs (Q2.3). Based on these analyses, we build a domain-pair classifier and evaluate its accuracy in different settings: within one system (dataset) and between different systems (Q3).

Eventually, this thesis leads to building a framework that can evaluate fitness of domains for cross-domain recommendation, select the best domain pairs for cross-domain recommendation, and perform cross-domain recommendation based on the selected source and target domains. An overview of such framework with the steps to achieve these goals is shown in Figure 1.

To explore the feasibility of this thesis, we performed some preliminary work that was reported in the proposal document. Specifically, we used one of the three proposed datasets (the Yelp academic) to run a pilot study. We implemented and ran the proposed CCA-based cross-domain algorithm (CD-CCA) on the Yelp dataset and compared the results with one of the proposed baseline single-domain algorithms (SVD++) and its cross-domain form (CD-SVD). We analyzed the results to find out if the selected cross-domain recommender results improve because of the approach or the added information; we performed partial analysis on a subset of the single-domain data characteristics and recommendation results; partly analyzed a subset of the cross-domain data statistics for the improvement of the picked cross-domain algorithms over the single-domain one; and looked at the nature of various domain pairs to get a deeper understanding of the analysis results.

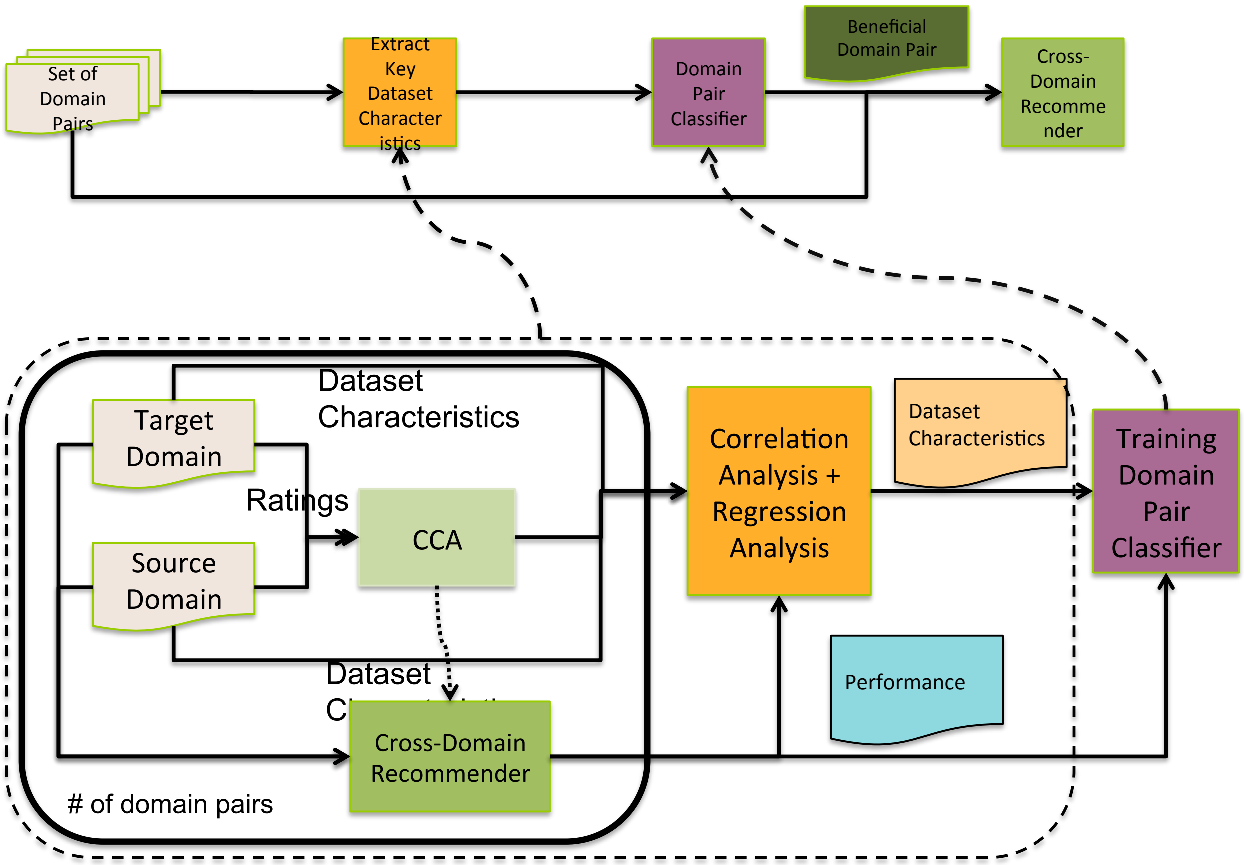


Figure 1: The proposed framework and tasks

## 1.3 ORGANIZATION OF THE THESIS

The rest of this dissertation is organized as follows. In Chapter 2, we review the closely related work in the literature to this dissertation and provide the background information required for the thesis. In Chapter 3, we introduce our proposed cross-domain algorithms and the baseline algorithms. In Chapter 4, we introduce the datasets that we are using in the experiments. Chapter 5 is dedicated to general experiments on and comparison of the proposed approaches and the baseline algorithms to answer research question Q1.1. In Chapter 6, we analyze the results for the cold-start situation to answer Q1.2. Chapter 7 includes an introduction to the dataset features we want to use in finding the appropriate domain pairs, the correlation analysis, and the regression analysis of these features with the error of single and cross-domain algorithms to answer Q2.1 and Q2.2. We then examine the domain pairs to understand if the correlation and regression results are in coordination with the intuitions about closely-related domain pairs (Q2.3). In Chapter 8 we introduce a domain classifier to find the best auxiliary domains and experiment with the classifier to examine its feasibility (Q3). Finally, in Chapter 9, we summarize the results of the research questions and list the contributions, limitations, delimitations, and possible extensions to this dissertation. Auxiliary materials are provided in Appendices A to C.

# 2.0 BACKGROUND AND RELATED WORK

In this chapter, we first introduce the notations that we will be using through this dissertation. Then, we briefly review the literature for recommender systems and various types of cross-domain recommenders. After that, we review a body of related collaborative filtering approaches that are not introduced as cross-domain, but are closely related to this ares. Eventually, we provide a summary of Canonical Correlation Analysis (CCA) and Large-Scale Canonical Correlation Analysis (L-CCA), as backgrounds for the proposed algorithms, and survey the previous application of CCA in the recommender systems fields of research.

## 2.1 NOTATION USED IN THIS DOCUMENT

In the subsequent chapters we will use the following notation:

- Matrices are shown in capital letters: $X$
- Vectors are shown in lowercase: $w$
- $y_{i,j}$ represents the value of $Y$ in row $i$ and column $j$
- $X^T$ shows the transpose of matrix $X$ and $w^T$ shows the transpose of vector $w$
- $X^{-1}$ represents the inverse of $X$
- $\hat{Y}$ shows the estimated values for $Y$
- $\tilde{Y}$ shows an incomplete matrix

The following is the definition of terms used in this proposal:

- **Domain** According to [34], domains can be categorized as system, data, and temporal domains. These categories represent, respectively, different datasets that a recommender system is built upon, various representation of user preferences (explicit or implicit), and various time points in which the data is gathered. In this proposal, we define domains based on the nature of items that exist in the domains, e.g., books vs. movies.

- **Target Domain** is the domain in which recommendations are performed. The recommended items are chosen from this domain.
- **Auxiliary or Source Domain** is the domain from which knowledge is transferred to help recommendations in the target domain.

## 2.2 SINGLE AND CROSS-DOMAIN RECOMMENDER SYSTEMS

Recommender (Recommendation) Systems aim to alleviate the information overload problem by helping users to select items from the provided item or information space. The first recommender system was introduced more than twenty years ago by Goldberg et al. to deal with the increasing amount of messages that users received by email [22]. This system utilized a technique called Collaborative Filtering (CF) to provide recommendations to a user based on past actions performed by herself and her nearest neighbors. At first, traditional recommender systems followed this trend of using collaborative filtering approaches [58]. After a while, rule-based, content-based, and hybrid approaches emerged to address various problems in recommender systems [47].

In rule-based recommender systems, decisions are made based on some rules that are extracted, either manually or automatically, from user profiles. In traditional cases, this method depended on knowledge engineering abilities of the system designers to build a suitable rule base for specific characteristics of the domain and market [57].

Content-based recommender systems, provide recommendations to users based on comparing items or products to the items that user had showed interest to. A user profile in these systems represents explanations of product contents that user chose before. These recommender systems usually rely on Information Retrieval techniques such as classification, clustering, and text analysis [49]. Unlike collaborative filtering methods, user profiles are created individually in these systems, only based on the items seen or rated by the user herself. We can name Letizia [39] and NewsWeeder [33] as first examples of content-based recommender systems.

Collaborative filtering-based recommender systems have achieved an acceptable success in e-commerce sites [23, 63]. These models usually include matching item ratings of current user (like rating on books, or movies) to similar users (close neighbors) to recommend items that are not yet seen/rated by this user.

Hybrid recommender systems were developed with the goal of solving the problems of content-based and collaborative filtering recommenders. These recommenders use various resources of information and combine both collaborative filtering and content-based methods [48, 38, 14, 2].

As the heterogeneity of data sources are increasing on the web, and due to the sparsity of data in each of these data sources, cross-domain recommendation has emerged as a research topic in the recent years. Although cross-domain recommendation is a recent field of study, it has gained increasing attention and is a promising way to develop new methods to improve recommendations, especially in a cold-start setting [60]. Cross-domain recommender systems aim to take advantage of information among related source (auxiliary) domains to recommend items in a target domain [20]. In some cases, the recommendation in the source and target domains can be performed simultaneously [36, 82], and in other cases, the recommendations are only delivered in the target domain [50].

A limitation of a significant number of works in cross-domain recommendation area is that they provide empirical results based on an artificial setting where either a single-domain dataset is subdivided into separate domains, e.g., separating the movie domain based on their genres [6, 11], or different user and items sets are used [52]. This is due to limited available cross-domain datasets. In order to overcome such a limitation, Dooms et al. mined twitter for structured rating tweets (such as "I liked X video on YouTube") for IMDB, Pandora, and Goodreads ratings, and YouTube likes [13], with the hopes of capturing some users with ratings in more than one system. Also, Zhang et al. developed a tool to record and analyze user browsing actions in web browsers, and provide browser-oriented cross-site recommendations [81].

Work on cross-domain recommendations includes collaborative filtering [21, 25, 28, 42, 60], content-based [7, 15, 29, 62, 68], and hybrid [18, 9, 74] approaches. In the following, we review the major work in each of these three categories and discuss the challenges in this field.

The research presented in this dissertation, and the proposed approaches, focus on cross-domain collaborative filtering approaches that share a common set of users between the source and target domains. To understand which features are important in finding the best-matched domain pairs, we design a set of comprehensive experiments and perform cross-domain collaborative filtering on multiple datasets. We use canonical correlation analysis as the main feature that can lead us to selection of best source domains for a target domain. To the best of our knowledge, we have the first comprehensive analysis for domain pair selection in cross-domain collaborative filtering.

### 2.2.1 Content-Based Cross-Domain Recommendations

There are few works in pure content-based cross-domain recommendation literature.

For example, Fernandez-Tobias et al. presented an ongoing research on a generic knowledge-based description framework built upon semantic networks in [19]. They automatically extracted information about two different domains, such as architecture and music, which are available in Linked Data repositories and performed weight spreading on the resulting concept graph to identify items in the target domain that were related to items of the source domain.

Low et al. used a hierarchical Bayesian model based on Latent Dirichlet Allocation and on latent side features for cross-property integration in Yahoo News and Yahoo Front Page [43].

Sahebi and Walker proposed a generic framework for content-based cross-domain recommendations in [62]. In this framework, an efficient method of feature augmentation is proposed to implement adaptation of domains. Instead of defining the notion of domain based on item descriptions, user-based domains are introduced. They applied their method in the job recommendation problem on Linkedin data.

Elkahky et al. proposed a content-based multi-view deep learning approach to cross-domain recommendations in [15]. They ran their experiments on Windows application recommendation, news recommendation, and movie/TV recommendation domains using search engine logs from Bing Web vertical, news article browsing history from Bing news vertical, app download logs from Windows AppStore, and movie/TV view logs from Xbox data plus public co-authorship data.

### 2.2.2 Collaborative Filtering Cross-Domain Recommenders

Cross-domain collaborative filtering aims to transfer user's rating pattern from source (auxiliary) domains to a target domain for the purpose of alleviating the sparsity problem and providing better target recommendations. Most of the work on cross-domain collaborative filtering has been either on manually picked, naturally close domains (e.g. movie and music) or on one domain that is randomly split into datasets considered as distinct domains.

As an example of recent work, Tirushi and Kuflik presented initial results of a work in progress that ranked and mapped between pairs of domains based on the ability to create recommendations in domain one using ratings of items from the other domain [70]. They collected $2,148$ Facebook profiles, which contained items (likes) in four domains: Music, Movies, TV shows, and Books. Their initial results, with cross-domain collaborative filtering on a joint space of domains, showed that there are differences between the source domains with respect to the quality of the recommendations.

Zhang et al. proposed MCF and MCF-LF methods that exploit the relationships between domains and perform multiple collaborative filtering tasks simultaneously [82]. They used a probabilistic framework which uses probabilistic matrix factorization to model the rating problem in each domain and allows the knowledge to be adaptively transferred across different domains by automatically learning a link function between domains. Their experiments were performed on MovieLens and Book Crossing datasets separately, each of which are divided randomly into five simulated domains. This approach does not need shared users or items between the domains.

In [26] constrained collective matrix factorization (CCMF) was proposed as an extension of collective matrix factorization ([66]) to iteratively factorize the rating matrices in source and target domain. The authors added a constraint on the user feature matrices for target domain and auxiliary domain. This approach assumes sharing users in the datasets and the experiments are on a simulated dataset sampled from the Netflix dataset and a real dataset crawled from Douban. Klami et al. also provided a method based on collective matrix factorization (CMF) [30]. This method allows each of the matrices to have a separate low-rank structure independent of the other matrices, as well as structures that are shared only by a subset of them. They tested the method on MovieLens and Flickr data.

Lu et al. proposed Selective Transfer Learning that transfers the data using a criterion based on empirical prediction error and its variance [45]. It extends Gaussian Probabilistic Latent Semantic Analysis (GPLSA) to Transferred Gaussian Probabilistic Latent Semantic Analysis (TGPLSA) model, then applies TGPLSA as base model over weighted instances for Selective Transfer Learning for Collaborative Filtering (STLCF). In this case, the approach needs either shared users or shared items.

Moreno et al. proposed a transfer learning technique (TALMUD) that extracts knowledge from multiple domains containing rich data (e.g., movies and music) and generates recommendations for a sparse target domain (e.g., games) [50]. The approach learns the degree of relatedness between different source domains and the target domain, without requiring overlapping users between domains. They tested their approach on Netflix, Jester, Music Loads, and Games Loads data.

Zhao et al. proposed a framework to construct entity correspondence between domains with limited shared user or items [83]. They used active learning to facilitate knowledge transfer across recommender systems based on Maximum-Margin Matrix Factorization. Their setting of source and target domains is as following: Netflix → Netflix, DoubanMovie → DoubanBook and Netflix → DoubanMovie.

Hu et al. proposed a generalized Cross Domain Triadic Factorization (CDTF) model over the triadic relation user-item-domain based on CP tensor decomposition [25]. They leveraged user explicit and implicit feedback respectively, along with a genetic algorithm based weight parameters tuning algorithm to trade off influence among domains optimally. They experimented on Amazon data (music CDs, DVDs and VHS video tape domains) and social network dataset provided by KDD Cup 2012 with 4 anonymous item domains.

Twin-Bridge Transfer Learning (TBT) proposed in [65] reduces the sparsity in target data by transferring knowledge from dense auxiliary data with either shared user or item sets and the similarity graphs of users and items constructed from the learned latent factors. The authors tested their approach on MovieLens10M and Epinions datasets separately with simulated domains created by random separation of datasets.

Wu et al. proposed a fusion multi-domain semantic topics and syntax classes model based on hidden Markov model with latent Dirichlet allocation (HMM-LDA) [77]. In every sub-domain, the model uses HMM-LDA to extract sub-domain topic and class features. Then, the fusion model combines the multiple sub-domain models to extract the whole domain features. They used MovieLens and Book-Crossing dataset (book and movie as source, movie as target) for their experiments. This approach does not require shared users or items.

Xin et al. proposed a nonlinear transfer learning model, and used the radial basis function (RBF) kernel to map user features of multiple sites [78]. This approach consists of two steps: first, the initial feature vectors for users/items in source and target domains are learned separately using probabilistic matrix factorization; then, a group of regression functions (using support vector machine) are used to map the user latent feature in the auxiliary domain to the user latent feature in the target domain. The kernel trick is used in this second step. In this approach the users should be shared in the domains. Douban (movies) and DianPing (restaurants) are the datasets the authors experimented on.

Loni et al. used factorization machines on Amazon data (books, music CDs, DVDs and video tapes) for cross-domain collaborative filtering [42].

Gao et al. [21] proposed a cluster-level based latent factor model for cross-domain recommendations. They based their optimization problems on a joint non-negative matrix tri-factorization. The assumption behind this factorization is that there is a common latent rating pattern across the two domains (in addition to domain-specific latent rating patterns) that drives the useful shared information. They tested their method on MovieLens, EachMovie, and Book-Crossing datasets.

Iwata and Takeuchi proposed a method based on matrix factorization, assuming that latent vectors in different domains are generated from a common Gaussian distribution with a full covariance matrix [27]. Neither users nor items were shared across domains. They tried their method on Movielens, EachMovie, Netflix, and Amazon review rating (Book, DVD, Electronics, Kitchen, Music and Video).

Liu et al. proposed the notion of Hyper-Structure Transfer (HST) and its model called the Minimal Orthogonal Tensor Approximation with Residuals (MOTAR) that transfers non-linearly correlated knowledge between domains [41]. This approach works on the domains with shared users. Movielens and DBLP (each citation is a rating, each category of MS research is a domain) are the datasets they have tested their approach on.

Mirbakhsh and Ling proposed cross-domain clustering-based matrix factorization on Amazon dataset (DVD, music, video, electronics, kitchen and housewares, and toys and games) and Epinions dataset (10 categories with the most observed ratings) in [46].

### 2.2.3 Hybrid Cross-Domain Recommendations

Many of the literature on transferring knowledge in recommender systems fall into the category of hybrid recommender systems. Specially since some literature consider hybrid-recommendations as cross-domain recommendation [34]. In these approaches two or more of the following types of information is used: user behavior data (such as user ratings, purchases, and logs), user content profile (such as user tags, the content of items consumed by users, or the semantic network behind them), and user social profile.

As an example of hybrid, cross-domain recommendation method, Acar et al. formulated the problem as a coupled matrix and tensor factorization (CMTF) problem, in which heterogeneous datasets are modeled by fitting outer-product models to higher-order tensors and matrices [1]. They proposed an optimization approach called CMTF-OPT, which is a gradient-based optimization approach for joint analysis of matrices and higher-order tensors. However, their data in their experiments is not cross-domain: they randomly generated matrices and tensors in a simulation.

Wang et al. proposed a Tag Transfer Learning (TTL) method that transfers tag topics instead of user-item rating patterns [74]. They used “MovieLens 10M Ratings, 100k Tags” data to perform this transformation on the movies domain only. Dong and Zhao analyzed the feasibility of tag-based cross domain rating prediction based on K-nearest neighbor model [12]. They reported the associative tag pairs of user preferences on items across domains on Douban dataset. This dataset consists of user rating and tag information on books, movies, and music in the Chinese Douban website.

Roy et al. recommended media (video) in social networks (twitter) with online stream LDA method (OSLDA) in [59]. They built a common topic space between the domain of social stream and video to do that.

In [32] Krohn-Grimberghe et al. added social data for recommendation by extending Bayesian personalized ranking (BPR) framework to the multi-relational case. They experimented on three social network datasets: Blogcatalog, Flickr, and YouTube; each dataset consists of relation between users and labels (target), and social relation between users and other users (auxiliary).

Enrich et al. proposed three tag-based rating prediction models using UserItemTags, UserItemRelTags, and ItemRelTags [16]. They experimented on MovieLens 10M and LibraryThing datasets.

Shapira et al. extracted users' favorite items and preferences in the domain of recommendation from Facebook content published by users on their personal pages [64]. They gathered the data about preferences related to other domains to allow cross-domain recommendation. They performed a field study with 95 subjects.

Chen et al. proposed a generalized cross domain framework that integrates social network information with collaborative-filtering data using tensor factorization [9]. They recommended users, tags, and items with topic based social regularization (FUSE) on the data from MovieLens dataset (source) and LibraryThing (target).

In [18], Fernandez-Tobias et al. adapted the gSVD++ algorithm to propose TagGSV++ that introduces a new set of latent variables, and enriches both user and item profiles with independent sets of tag factors on MovieLens and LibraryThing data.

Co-Citation Selection (CCS) [69], was proposed based on collaborative filtering on co-citation networks, in which neighboring papers were selected and weighted into publication citation prediction.

### 2.2.4 Shared Data in Cross-Domain Recommenders

Many cross-domain algorithms assume that the source and target domains share at least one of the user or item spaces [45, 65, 78, 41]. In some cases, having partial shared user or item sets is sufficient [30, 83]. In other cross-domain approaches, the assumption is that no shared users or items are needed in the domains [36, 82, 77, 27].

However, there is some controversial literature regarding the no-sharing cross-domain approaches. Cremonesi and Quadrana provide empirical evidence in [10] that CBT (code book transfer, one of the pioneer no-sharing cross-domain approaches presented in [35]) improves the accuracy of recommendations without transferring knowledge from source to target domain. They show that the injection of the codebook in the target domain is equivalent to a two-matrix factorization algorithm without transfer of knowledge from the source domain and the increase of accuracy measured in this approach is due to a pitfall in the evaluation procedure.

### 2.2.5 Selecting the Best Auxiliary Domain

Although a considerable amount of work has recently been done on cross-domain recommendations, most of the research either assumes that the selected domains are related to each other or the experiments are conducted on simulated domains, generated from one domain.

In some of the previous works, such as [82, 45], the researchers focused on transferring some of the records from the source domain to the target domain based on some criterion or weighted transfer. However, they have not studied the general relatedness of the source and target domains.

In addition to our preliminary work, the only other work in this area is by Yi et al. [80]. They select auxiliary domains in movie recommendation (based on movie genres). They conclude that Kullback-Leibler (KL) divergence between non-overlapping user ratings and the number of overlapping users between target and auxiliary domains are indications of choosing a helpful domain. This study is limited to the MovieLens 1M dataset with simulated domains.

### 2.2.6 Related Collaborative Filtering Approaches

In some of the collaborative filtering approaches, transfer learning within one domain is used to alleviate the sparsity problem or add extra information for the recommendation. Although this is not considered cross-domain recommendation, these works worth noting as related work.

For example, Pan et al. proposed transfer by integrative factorization (TIF), that uses auxiliary uncertain ratings (a rating distribution as a rating spectrum involving uncertainty instead of an accurate point-wise score) to improve the performance of recommendation [53]. They integrated auxiliary data of uncertain ratings as additional constraints in the target matrix factorization problem, and learned an expected rating value for each uncertain rating. The experiments were not on different domains and the authors split the experimental data randomly and disturbed some of the ratings as the auxiliary rating profile. The experiments were done on MovieLens and Netflix datasets.

In their next paper, Pan et al. used additional auxiliary data in the form of binary ratings, transferring knowledge to a target numerical rating matrix of the same domain [54]. Their framework, Transfer by Collective Factorization (TCF), constructs a shared latent space collectively and learns the data-dependent effect separately. They experimented on Moviepilot rating data and Netflix data. Li et al. defined the collaborative filtering domains as a 2-D site-time coordinate system, on which multiple successive time-slices, can share group-level rating patterns [37]. They developed a generative model: ratings over site-time (ROST) and used MovieLens dataset to run their experiments.

Parimi and Caragea proposed a method based on a regularized latent factor model, using implicit feedback [55]. This approach can handle variable user overlap. The authors have tried the approach on last.fm (artist, friend, and tag domains) and DBLP (co-author, conference and reference domains) datasets.

In addition to the transfer learning approaches, context-aware recommenders are also close to cross-domain recommendation research field. For example, Liu et al. presented the Contextual Operating Tensor (COT) model, that represents the common semantic effects of contexts as a contextual operating tensor and represents a context as a latent vector [40].

They assumed that context combinations can operate the latent characteristics of entities. They experimented on Food (virtuality and hunger contexts), Adom (movie data companion, when, release, rec, and where as contexts) , and Movielens-1M (hour and day timestamp as context) datasets.

Another set of related collaborative filtering approaches aim to improve the recommendations by introducing external rating information, in an aggregated format, to the target domain. For example, Adomavicius and Tuzhilin proposed using aggregated user ratings in a hierarchical recommender system [3]. Umyarov and Tuzhilin proposed to use aggregated ratings from various user segments, in the form of parameter constraints, to improve recommendations [71]. They later introduced a general class of methods that combined external aggregate information, in the form of average and variance of the ratings, along with individual ratings [72]. They experimented on MovieLens and Netflix data to show that the aggregate average ratings are good enough to improve the recommendations. Umyarov and Tuzhilin theoretically proved in [73] that adding the aggregate rating information results in better predictions of unknown ratings and empirically showed that it alleviates the cold-start problem. The proof is based on the idea that adding the aggregate ratings reduces the variance of estimated ratings, and thus, leads to less error.

## 2.3 CANONICAL CORRELATION ANALYSIS

We use canonical correlation analysis as a main building block of our proposed algorithm and as an important factor in the domain-pair selection experiments and analysis. In the following sections, we review regularized CCA and large-scale CCA to provide a background for the remaining chapters in the dissertation.

### 2.3.1 Regularized CCA

Canonical correlation analysis (CCA) is a multivariate statistical model that studies the interrelationships among sets of multiple dependent variables and multiple independent variables. It is the most generalized member of the family of multivariate statistical techniques [24]. It is related to factor analysis in the sense that it creates composites of variables, and is related to discriminant analysis in finding independent dimensions for each variable set. The goal of this analysis is to produce the maximum correlation between the dimensions. As a result, canonical correlation finds the optimum structure or dimensionality of each variable set that maximizes the relationship between independent and dependent variable sets.

In other words, if we have $X \in \mathbb{R}^{m\times n}$ and $Y \in \mathbb{R}^{p\times n}$, CCA finds two projection vectors $w_x \in \mathbb{R}^m$ and $w_y \in \mathbb{R}^p$ that maximize the correlation coefficient:

$$\rho = \frac{w_x^T XY^T w_y}{\sqrt{(w_x^T XX^T w_x)(w_y^T YY^T w_y)}} \tag{2.1}$$

Since Equation 2.1 is not affected by re-scaling of $w_x$ and $w_y$ (the multiplication of these vectors by a constant $\alpha$ does not change the value of $\rho$), we can maximize $\rho$ as follows.

$$\begin{aligned} &\max_{w_x, w_y} w_x^T XY^T w_y \\ &\text{subject to } w_x^T XX^T w_x = 1, w_y^T YY^T w_y = 1 \end{aligned} \tag{2.2}$$

It can be shown that solving Equation 2.2 is equivalent to finding the eigenvectors of top eigenvalues of the generalized eigenvalue problem in Equation 2.3, in which $\eta$ is the eigenvalue that corresponds to the eigenvector $w_x$.

$$XY^T(YY^T)^{-1}YX^T w_x = \eta XX^T w_x \tag{2.3}$$

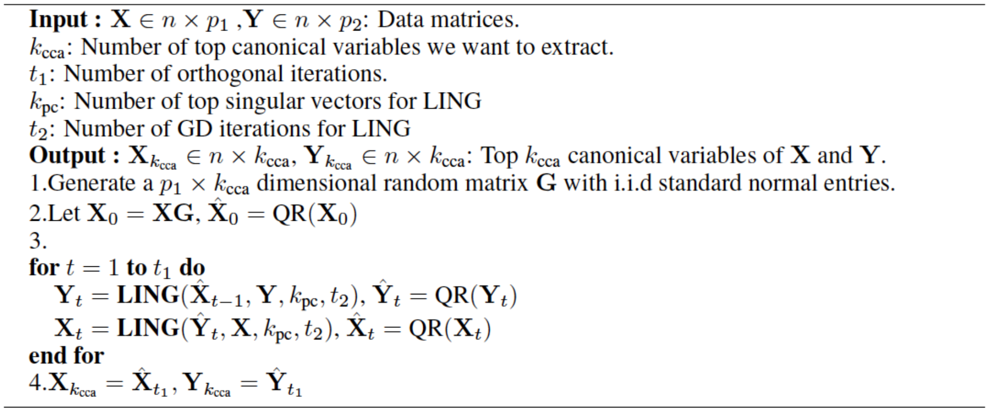
To compute multiple projection vectors, we can solve the optimization problem in Equation 2.4, in which matrix $W$ consists of multiple projection vectors.

$$\begin{aligned} &\max_{W} Trace(W^T XY^T (YY^T)^{-1} YX^T W) \\ &\text{subject to } W^T XX^T W = I \end{aligned} \tag{2.4}$$

To avoid the over-fitting of $\rho$ and the singularity of $XX^T$, a term $\lambda I$ is added to Equation 2.3. We have the constraint $\lambda > 0$ in this regularization term. Eventually, the regularized CCA attempts to solve the generalized eigenvalue problem in Equation 2.5.

$$XY^T(YY^T)^{-1}YX^T w_x = \eta(XX^T + \lambda I)w_x \tag{2.5}$$

Sun et al. solve the regularized CCA problem, using a least squares formulation of it, with the Least Angle Regression algorithm [67].



```
**Input :** $\mathbf{X} \in n \times p_1$ ,$\mathbf{Y} \in n \times p_2$: Data matrices.
$k_{\text{cca}}$: Number of top canonical variables we want to extract.
$t_1$: Number of orthogonal iterations.
$k_{\text{pc}}$: Number of top singular vectors for LING
$t_2$: Number of GD iterations for LING
**Output :** $\mathbf{X}_{k_{\text{cca}}} \in n \times k_{\text{cca}}$, $\mathbf{Y}_{k_{\text{cca}}} \in n \times k_{\text{cca}}$: Top $k_{\text{cca}}$ canonical variables of $\mathbf{X}$ and $\mathbf{Y}$.
1.Generate a $p_1 \times k_{\text{cca}}$ dimensional random matrix $\mathbf{G}$ with i.i.d standard normal entries.
2.Let $\mathbf{X}_0 = \mathbf{XG}$, $\hat{\mathbf{X}}_0 = \text{QR}(\mathbf{X}_0)$
3.
**for** $t = 1$ **to** $t_1$ **do**
    $\mathbf{Y}_t = \textbf{LING}(\hat{\mathbf{X}}_{t-1}, \mathbf{Y}, k_{\text{pc}}, t_2)$, $\hat{\mathbf{Y}}_t = \text{QR}(\mathbf{Y}_t)$
    $\mathbf{X}_t = \textbf{LING}(\hat{\mathbf{Y}}_t, \mathbf{X}, k_{\text{pc}}, t_2)$, $\hat{\mathbf{X}}_t = \text{QR}(\mathbf{X}_t)$
**end for**
4.$\mathbf{X}_{k_{\text{cca}}} = \hat{\mathbf{X}}_{t_1}$, $\mathbf{Y}_{k_{\text{cca}}} = \hat{\mathbf{Y}}_{t_1}$
```

Figure 2: L-CCA algorithm as presented in [44]

### 2.3.2 Large-Scale CCA

Calculating CCA can be very resource-consuming especially in the traditional approaches that should calculate QR-decompositions or singular value decomposition of large data matrices. To avoid these time and memory consuming operations, Lu and Foster developed an iterative algorithm that can approximate CCA on very large datasets [44]. They establish an error analysis for the case of having finite number of iterations in the algorithm and prove that the algorithm converges to the real value of CCA in case of infinite iterations.

This approach relies on LING, a gradient-based least squares algorithm that can work on large-scale matrices. As we have seen in the previous section, CCA can be computed as an iterative least squares problem. So, to compute CCA in L-CCA, first a projection of one of the data matrices on a randomly-generated small matrix is generated, to reduce the size of the matrix. Then, a QR-decomposition of this smaller matrix is calculated. After that, the CCA is calculated iteratively, by applying LING on the reduced-sized QR-decompositions of the original data matrices, in each iteration. Every time after running LING, a QR-decomposition is calculated for numerical stability. A summary of this algorithm that is presented in [44] is shown in Figure 2.

The LING algorithm relies on the intuition that the projection of independent variables on the least square estimates, can be divided (column-wise) into two smaller orthogonal components, each of which is related to the top (or bottom) singular vectors of the data. Then, it computes the first orthogonal component using randomized SVD, and the second one using gradient descent algorithm.

To be more specific, considering the least-squares problem of $Y = X\beta$, then $X\beta^* = X(X^TX)^{-1}X^TY$ is the projection of $Y$ into column space of $X$[1]. Calculating $(X^TX)^{-1}$ takes a long time for a large $X_{n\times p}$. To calculate $X\beta^*$ without the need of calculating $(X^TX)^{-1}$, Lu and Foster rely on splitting the singular vectors of $X$.

If $U_1$ is the top $k_{pc}$ singular vectors of $X$, and $U_2$ is the remaining $p - k_{pc}$ singular vectors, $X\beta^*$ can be divided into two orthogonal vectors as in Equation 2.6. Then, they calculate the first term using randomized SVD, since $K_{pc} < p$. Let $Y_r = Y - U_1U_1^TY$. Then the second term can be calculated using gradient descent for $Y_r = X\beta_r$.

$$X\beta^* = U_1U_1^TY + U_2U_2^TY \tag{2.6}$$

Lu and Foster provide an error bound for the Ling algorithm and an error bound for L-CCA based on that in [44].

### 2.3.3 CCA in Recommender Systems

CCA has been used in different literature for the single-domain recommenders with various resources or to find the correlation between the content (such as text or image) of the resources in cross-domain recommender systems. To the best of our knowledge, it has not yet been used in a pure, rating-based, cross-domain collaborative filtering setting. For example, in the area of recommender systems, Faridani has used CCA to predict hotel ratings from textual comments of the hotels and their sentiment analysis [17]. Elkahky et al. use CCA as a baseline user modeling approach for their proposed recommendation system in [15]. They provide content-based cross-domain recommendations in the domains of apps, news, movies, and TV shows using a multi-view deep learning model. In [51], Ohkushi has used Kernel CCA in context-aware setting to find the relationship between music pieces and human motion to recommend music to users. Yang et al. [79] have proposed a feature learning algorithm that

[1] $\beta^*$ is the estimate for $\beta$

uses CCA for inferring features of semantic information in the data. However, Yang et al. have not yet used their model in recommender systems.

# 3.0 CCA-BASED CROSS-DOMAIN ALGORITHMS

In this chapter, we propose cross-domain collaborative filtering algorithms based on regularized CCA and large-scale CCA. We base our approaches on canonical correlation analysis, because we hypothesize that this tool can lead us to a better understanding of the relationship between the domain pairs and thus, a better cross-domain recommendation results. The proposed approaches can be categorized into cross-domain collaborative filtering methods that require a shared set of users between the source and target domains. These approaches are used in finding the answers to the research questions presented in this thesis. In the last sections of this chapter, we introduce the baseline algorithms that we use in the experiments to compare the results with the proposed algorithms.

## 3.1 CD-CCA

As explained in Section 2.3.1, CCA evaluates the latent linear correlations between two sets of variables. To draw an analogy between CCA and cross-domain recommender, we suppose that there are $n$ common users between the source and target domains. We consider the source (auxiliary) domain in cross-domain recommender as the independent variable set $X$ (with $n$ users and $m$ items), and the target domain as the dependent variable set $Y$ (with $n$ users and $p$ items). Note that here we are working on $m \times n$ and $p \times n$ item-user matrices, as opposed to the usual user-item matrices in collaborative filtering. The value $\rho$ in Equation 2.1 shows the maximum canonical correlation that can be achieved by rotating the $X$ and $Y$ spaces in direction of $w_x$ and $w_y$, respectively. In other words, CCA calculates the components of each domain, that are consisted of sets of items from each of the domains,

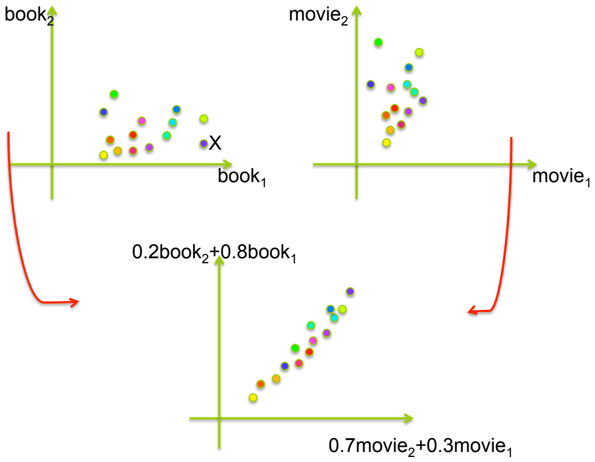


Figure 3: A toy example of CCA in cross-domain recommender system setting

which are most similar to each other based on user rating behavior. Also, it determines how much the two components are correlated to one another.

As an example this analogy, we can look at Figure 3. In this example, we assume that we have "books" (upper left-hand side) and "movies" (upper right-hand side) domains. We assume that each domain has two items in it: "book1" and "book2" are a set of dependent variables in the "books" domain and "movie1" and "movie2" are a set of independent variables in the "movies" domain. The axes show user ratings on these items in the domains. Each user in each domain is represented by one dot. Users are separated by dots with different colors: the purple dot in the "books" domain shows the rating of the same user as the purple dot in the "movies" domain. For example, the user marked by "X" has a high rating on "book1" and a low rating on "book2". CCA finds the components of each of the "books" and "movies" domains so that the correlation between user ratings, represented in these components, are maximized (the lower picture). These components are linear combinations of items in the two domains. In this example, the component found by CCA in the "books" domain is a linear combination of 0.2 of rating values on "book1" and 0.8 of rating values on "book2".

As a result, if we know the ratings in the source domain $X$ and ratings in the target domain $Y$, we can find the $w_x$ and $w_y$ that maximize the canonical correlation between $X$ and $Y$. In other words, with the projections vectors $w_x$ and $w_y$, we know how the ratings of a combination of items in the source domain affect the ratings of an item in the target domain. Consequently, after adding the user ratings of the source domain $X$, we can understand how all of the ratings of a user in the source domain affect the same user's ratings in the target domain. Eventually, we can estimate the ratings of users in the target domain $\hat{Y}$ by using the projection vectors, the source domain ratings, and the canonical correlation value [75]. The calculation of estimated rating ($\hat{Y}$) is shown in Equation 3.1. Thus far, this approach only focuses on the first canonical component (projection vectors) that maximize the correlation ($\rho$ or R-statistic). There are other components between the domains that can indicate different projection vectors and correlations (R-Statistics) for each pair of them. In this case of multiple projections, the estimated rating matrix $\hat{Y}$ is calculated as in Equation 3.2. Here, if we assume that $c$ pairs of projection vectors are calculated, $P$ is a diagonal $c \times c$ matrix, in which the diagonal elements are $\rho$s for each canonical component; $W_x$ is a $m \times c$ matrix consisted of $c$ projection vectors of size $m \times 1$; and $W_y$ is a $p \times c$ matrix of $c$ projection vectors of size $p \in 1$.

$$\hat{Y} = w_y \rho w_x^T X \tag{3.1}$$

$$\hat{Y} = W_y P W_x^T X \tag{3.2}$$

If the target rating matrix is incomplete and has some missing values ($\tilde{Y}$), we can estimate $W_x$ and $W_y$ ($\hat{W_x}$ and $\hat{W_y}$) by calculating the canonical correlations between the source rating matrix $X$ and incomplete target matrix $\tilde{Y}$. Then, we can use the estimated projection vectors $\hat{w_x}$ and $\hat{w_y}$ to estimate a complete rating matrix $\hat{Y}$. More specifically, if we want to predict the unknown rating of user $i$ on item $j$ in the incomplete target domain ($\hat{y}_{j,i}$), we follow Equation 3.3 after finding $\hat{W_x}$ and $\hat{W_y}$ on matrices $X$ and $\tilde{Y}$. Here, $X_{k,i}$ is the rating of user $i$ on item $k$; $\hat{W}_{y_{j,l}}$ refers to the target projection element for the item $j$ and component $l$; and $\hat{W}_{X_{k,l}}$ is the source projection element for the item $k$ and component $l$.

$$\hat{y}_{j,i} = \Sigma_{l=1}^{c} \hat{W}_{y_{j,l}} P_{l,l} \Sigma_{k=1}^{m} \hat{W}_{X_{k,l}} X_{k,i} \tag{3.3}$$

As an abbreviation, we use the name CD-CCA for this CCA-based cross-domain recommender. The process of mapping between source and target domains in CD-CCA is shown in Figure 4.

One of the problems with this CD-CCA algorithm is its scalability. Calculating matrix multiplications in large scale and dense format can be difficult in terms of both memory and processing requirements. As a result, in the next section, we propose a new algorithm (called CD-LCCA) that uses large-scale CCA [44] to alleviate this problem.

## 3.2 CD-LCCA

As we have seen in section 2.3.2, large scale CCA finds a lower-dimensional representation of each of the input matrices and then calculates the canonical correlation analysis between these two matrices. To base our cross-domain recommender algorithm on LCCA, suppose that we have a $n \times m$ source domain rating matrix $X$ and a $n \times p$ target domain rating matrix $Y$. Here, $n$ represents the number of shared users between the source and target domains; $m$ shows the number of items in the source domain; and $p$ shows the number of items in the target domain. Suppose that $X_c$ $(n \times x_c)$ is the lower dimensional matrix that represents the source domain rating matrix $X$, and $Y_c$ $(n \times y_c)$ is the lower dimensional matrix that represents the target rating matrix $Y$ in the LCCA algorithm. Then, if we calculate the canonical correlations between $X_c$ and $Y_c$, we will have $X_c W_{x_c}$ $(n \times k_{cca})$ and $Y_c W_{y_c}$ $(n \times k_{cca})$ as canonical variates and $P$ $(k_{cca} \times k_{cca})$ as the canonical correlation between these variates. Thus, we can map $X_c$ to $Y_c$ (and vice versa) based on these canonical correlations and variates. For example, $Y_c$ can be achieved using Equation 3.4.

$$Y_c = X_c W_{x_c} P W_{y_c}^T \tag{3.4}$$

Although Equation 3.4 maps the source and target domains by building a relationship between their lower dimensional representations ($X_c$ and $Y_c$), we need to be able to map the original source and target matrices ($X$ and $Y$) to be able to estimate user ratings in them. To build a relationship between the original source and target domain matrices, we first look

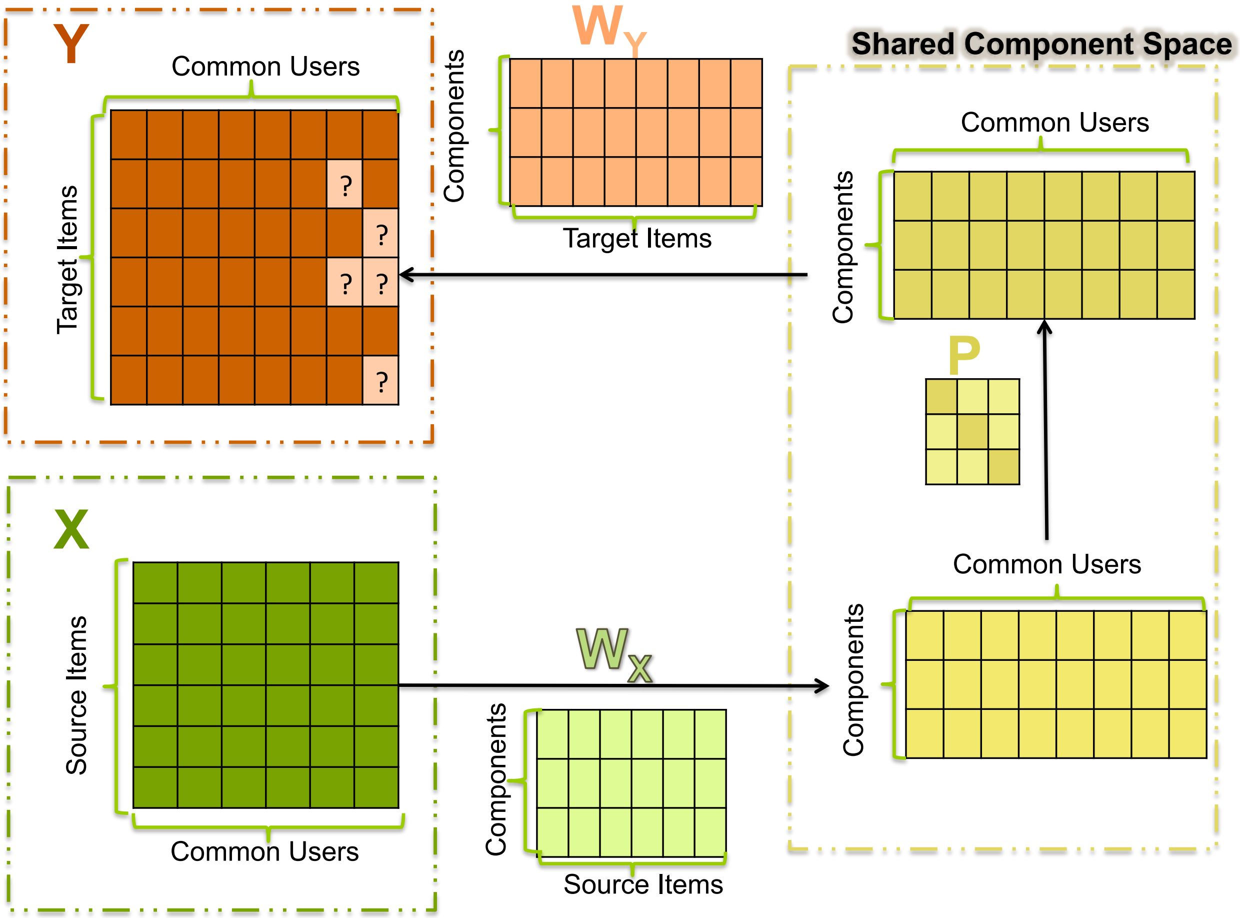


Figure 4: Mapping between source and target domains in CD-CCA

at the relationship between each domain matrix and its lower dimensional representation. Considering the source domain matrix ($X$), we build $X_c$ in the first step of LCCA by solving an iterative least square problem, having a QR-decomposition in each iteration. Although we loose the mapping information between $X$ and $X_c$ in this iterative process, having both $X$ and final $X_c$ matrices, we can restore the mapping that happens between them. Since $X_c$ is a lower dimensional projection of $X$, we can write their relationship as in Equation 3.5. Here, $M$ is a $m \times c_x$ mapping that projects the $n \times m$ matrix $X$ into the $n \times c_x$ matrix $X_c$.

$$X_c = XM \tag{3.5}$$

Consequently, we can find the mapping $M$ by the inverse relationship between $X$ and $X_c$ using Equation 3.6.

$$M = X^{-1}X_c \tag{3.6}$$

The same can be applied to find the mapping of target rating matrices $Y$ and its lower-dimensional representation $Y_c$ (Equation 3.7).

$$N = Y^{-1}Y_c \tag{3.7}$$

So, we can also rebuild $Y$ based on $N$ and $Y_c$ ($Y = Y_cN^{-1}$). Combining Equations 3.7, 3.6, and 3.4, we can now map between the original source and target rating matrices as presented in Equation 3.8 and have an estimation of user ratings in the target domain ($\hat{Y}$).

$$\hat{Y} = XMW_{x_c}PW_{y_c}^{-1}N^{-1} \tag{3.8}$$

As a result, if we would like to estimate the rating of user $i$ on item $j$, we can use:

$$\hat{y}_{i,j} = \Sigma_{q=1}^{m} X_{i,q} \Sigma_{o=1}^{c_x} M_{q,o} \Sigma_{l=1}^{k_{cca}} W_{x_{c_{o,l}}} P_{l,l} \Sigma_{r=1}^{c_y} W_{y_{c_{l,r}}} N_{r,j}^{-1} \tag{3.9}$$

Since $X$ and $Y$ matrices are sparse, we take advantage of this property in Matlab implementation to reduce the memory requirements. Having $U = X_cW_{x_c}$ as an output of Matlab's "canoncor" function, we can skip this multiplication. Calculating $N$ is fast and efficient using the "mldivide" function[1]. We can thus calculate $\hat{Y}$ using Equation 3.10.

$$\hat{Y} = UPW{y_c}^{-1}N^{-1} \tag{3.10}$$

Note that $U$, $W_{yc}$, and $N$ (and thus the calculated matrix $\hat{Y}$) are dense matrices. The density of $U$, $W_{yc}$, and $N$ is not problematic in terms of memory because they are all low-dimensional matrices (compared to $\hat{Y}$). To be more memory-efficient in calculating $\hat{Y}$, we first break Equation 3.10 to a multiplication of two dense matrices ($A$ and $B$) in lower dimensions. Then, we strip away the unnecessary values from these two matrices and transform them to the Matlab's sparse format. By multiplying these new sparse matrices, we will achieve a sparse estimation of $Y$.

To this end, we compute $A = W{y_c}^{-1}N^{-1}$ (which is a $k_{cca} \times p$ matrix), and $B = UP$ (which is a $n \times k_{cca}$ matrix). Since we need to calculate the rating values only for test users and test items, not all rows and columns of $A$ and $B$ are required. If $S \subseteq \{1..n\}$ shows the set of test users, and $I \subseteq \{1..p\}$ represents the set of target items we need to estimated user ratings on, we can build the sparse sub-matrix of $A$ ($\tilde{A}$) and the sparse sub-matrix of $B$ ($\tilde{B}$) as following:

$$\tilde{A}_{i,:} = \begin{cases} A_{i,:}, & \text{if } i \in S \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

$$\tilde{B}_{i,:} = \begin{cases} B_{:,i}, & \text{if } i \in I \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

Eventually, we will have[2]:

$$\hat{Y} = \tilde{A}\tilde{B} \tag{3.13}$$

[1] If the source matrix size is too big and "mldivide" function takes too long, we take advantage of the column-wise independence of "mldivide" (or the fact that $[A|B]^{-1}C = [A^{-1}C|B^{-1}C]$). Thus, we separate the source matrix into multiple smaller matrices, using column-wise partitioning. Then, we apply the "mldivide" function on each of these matrices and eventually join the results together. In other words: $N = Y^{-1}Y_c = ([Y_1|Y_2]^{-1}Y_c) = [Y_1^{-1}Yc|Y_2^{-1}Y_c]$.

[2] The matrix multiplication tricks explained here are for using Matlab. Since Matlab is more efficient in working with matrices, compared to having "for" loops, we use these tricks. If another language is used for implementing this algorithm, we can use Equation 3.9 with looping over $\Sigma$s to have a fast implementation of the algorithm.

## 3.3 BASELINE ALGORITHMS

To study the improvement of cross-domain algorithms over single-domain ones and to select the best domain matches, we need to compare cross-domain algorithms with the single-domain ones. Also, to study the performance of the proposed algorithm, we need to compare and contrast it with other state-of-the-art cross-domain algorithms. Additionally, we will study if the improvements achieved using cross-domain recommendations are because of the additional data provided to them, or because of the algorithm itself. To do this, we use both domains' data as an input to the single-domain algorithm and compare it with other cross-domain baselines and the single-domain algorithm with target domain's data.

As baseline algorithms, in addition to CD-CCA, we run the `SVD++` algorithm [31][3], Rating-Matrix Generative Model (`RMGM`) [36], and Collective Matrix Factorization (`CMF`) [66] as some of the previous work compared their results to these algorithms.

### 3.3.1 SVD++

`SVD++`[31] is a single-domain algorithm, based on matrix factorization. In this algorithm, the rating matrix is decomposed into two smaller matrices: user-factor matrix ($Q$) and item-factor matrix ($P$). This decomposition is shown in Equation 3.14. Here, $\hat{r}_{ui}$ represents the estimated rating of user $u$ on item $i$; $q_i$ shows the user vector, e.g. the row representing user in the user-factor matrix; and $p_u$ shows the item vector, e.g. the row representing the item in the item-factor matrix. The user-factor matrix can be interpreted as user interests in the discovered factors and the item-factor matrix shows how much each item belongs to each factor.

$$\hat{r}_{ui} = q_i^T p_u \tag{3.14}$$

This decomposition is solved as an optimization problem. The goal is to minimize the Root Mean Squared Error (RMSE) of predicted vs. actual user ratings. Since users can have a bias in their ratings (e.g. a user may rate most of the products higher than average),

[3] Using GraphChi Software (http://graphchi.org)

this algorithm corrects for the user bias using user averages ($b_u$). Similarly, item-bias ($b_i$) and general bias ($\mu$) are added to the optimization problem. In addition to these biases, there is implicit information regarding the items that users choose to rate, regardless of their rating value. To account for this information, a second set of item factors ($y_j$) is added to characterize users based on the set of items they have rated. Since users do not rate all of the available items, the actual user-rating matrix is sparse. Thus, SVD++ only uses the observed ratings of each user in estimating the $P$ and $Q$ matrices. In order to achieve this, a set $R_u$ that represents the items rated by users is used.

Eventually, Equation 3.15 shows the final formulation for estimating user $u$'s rating on item $i$.

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T(p_u + |R(u)|^{-\frac{1}{2}}\Sigma_{j\in R(u)}y_j) \tag{3.15}$$

**3.3.1.1 SD-SVD and CD-SVD** In this thesis, we use SVD++ in two modes: single-domain and cross-domain. In the single domain mode (SD-SVD), we only use the target domain ratings to predict the test user ratings in the target domain. This serves as a baseline for studying if extra domain information helps in achieving better recommendations. For the cross-domain mode (CD-SVD), we add the source domain item space to the target domain item space, as if the two categories are coming from the same (single) domain. This is done by concatenating the source and target domain rating matrices. We run the SVD++ algorithm on the joined domain matrices. By studying this algorithm as a baseline for other cross-domain algorithms, we can study if the improvement of cross-domain recommenders is because of the added data of the target domain, or because of the approach to integrate this data. Figure 5 shows the cross-domain setup for the SVD++ algorithm.

### 3.3.2 Rating Matrix Generative Model (RMGM)

Rating Matrix Generative Model (RMGM) [36] is a generative model that creates a shared cluster-level matrix, which represents the relatedness across multiple rating matrices. In this model, rating of a user on an item is drawn from both a user-item joint mixture model and

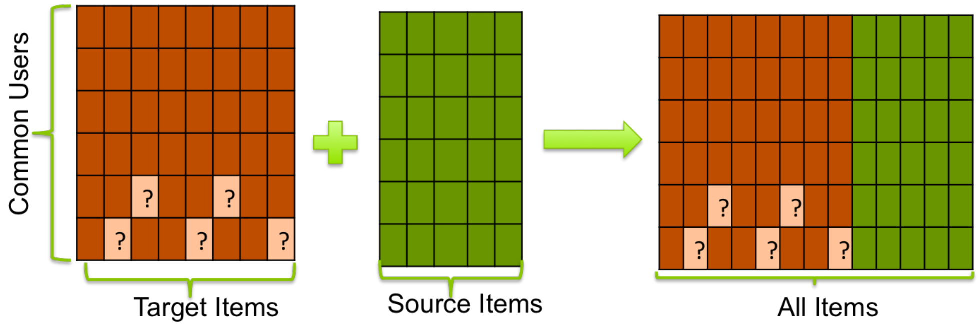


Figure 5: Cross-Domain Setup for SVD++

a corresponding ratings from this cluster-level rating model. An adaptation of Expectation Maximization (EM) algorithm is used for training the model.

This algorithm works for categorical ratings and requires to have the number of user and item clusters as its input. Since it is a generative model, it can work for unobserved or new users. However, since it relies on the shared cluster of users and items, it has problems when there is a high skewness in the ratings. When having high skewness in the ratings, users usually only rate the most popular items. In this case, the groupings of items and users based on ratings will not result in clear clusters. As a result, the error of this algorithm increases in these cases. The same happens when there is an extreme case of sparsity, or an extreme case of cold-start. RMGM will not be able to generate clear groupings of extreme cold-start users and this leads to a larger error. Additionally, RMGM does not require shared users or shared items between the source and target domains.

### 3.3.3 Collective Matrix Factorization (CMF)

Collective Matrix Factorization (CMF) was proposed by Singh and Gordon in 2008 [66] for learning multiple relationships in the same domain (e.g. predicting the movie ratings of users in the Netflix dataset, considering the genre relationship of movies from the Internet Movie DataBase). Although it was not proposed as a cross-domain algorithm, it has been used as a baseline in many cross-domain recommendation studies [26, 30].

This approach simultaneously factors several matrices by sharing parameters, learned for common entities in the relations, among factors. More specifically, if there are two data matrices $X$ and $Y$, it factorizes these matrices into factors $U$, $V$, and $Z$, such that $X \approx f_1(UV^T)$ and $Y \approx f_2(VZ^T)$. To find the appropriate factors, CMF uses the $L_2$ losses, once for approximating $X$ with $U$ and $V$, and once for approximating $Y$ with $V$ and $Z$; and minimizes the weighted average loss over these two losses. This optimization is done by alternating projection algorithm, updating for one of the factors at each time using a Newton-Raphson step.

In the cross-domain interpretation of this algorithm, the assumption is that $V$ represents the shared user factors, and it is shared between the source and target domains. It requires the source and target domains to have the same set of users.

CMF has problem in fitting accurate item factors for fat source and target domain rating matrices. This algorithm tries to represent user preferences, and items in both source and target domains via the same number of factors. In other words, the number of factors in $V$ and $Z$ are the same and equal to the number of user factors in $U$. Thus, if the number of users is much smaller than the number of items in the source or target domains, the number of factors to represent the tastes of a small set of users may not be enough to represent the large number of items. Also, it the number of items in the source and target domains are very different, the number of factors used to represent them may be very different. In this case, CMF has to either sacrifice the representation accuracy of one domain and use a small number of factors for both domains, or has to incorporate more factors and increase the risk of over-fitting for the domain with smaller number of items.

## 3.4 SUMMARY

In this chapter we introduced our proposed algorithms CD-CCA and CD-LCCA. We also presented a brief introduction to the baseline algorithms used in this thesis and explained their strengths and weaknesses.

CD-CCA and CD-LCCA are both built on canonical correlation analysis that finds the linear interrelationships between a set of dependent and a set of independent variables. Many factorization-based recommender algorithms, including the proposed and baseline algorithms in this dissertation, work based on assuming linear relationships between item vectors or domains. Although using only linear relationships is a limitation for these algorithms, they have shown reasonable performances in recommender systems literature. However, there are ways to incorporate non-linear relationships in the proposed algorithms. One of which is using kernel-CCA [4] instead of linear CCA to find the relationship between the source and target domains. However, using kernel-CCA for recommendation requires a mapping from the source domain space to the kernel space and from the kernel space back to the target space.

Table 1 shows a summary of the proposed and baseline algorithms with some of their properties. While CD-CCA, CD-LCCA, CMF, and RMGM are all algorithms that are designed to map between two or more domains, CD-SVD is a single-domain algorithm that receives a union of source and target domain ratings as its input. CMF, CD-SVD, and SD-SVD are all based on matrix factorization models. CD-CCA is similar to these algorithms in the sense that it maps the user-item rating matrices into a lower-dimensional space. The same is true for RMGM.

RMGM does not require any shared users or items between the domains, while other cross-domain algorithms need to have shared users. Also, RMGM works on categorical input data. So, it cannot process the Supermarket dataset directly as its input. The rest of the algorithms assume to have a continuous input space. The output of these algorithms are also continuous values. Since many recommender system datasets have nominal ratings of users on items, this can be a limitation of these algorithms.

As we will see in the next chapters of this dissertation, these algorithms are different in their running times. CD-CCA, CD-LCCA, and SVD++ are among the fastest algorithms. RMGM and CMF are slower. Also, as discussed in this chapter, each of these algorithms have their own limitations. For example, RMGM performs poorly in cases of extreme cold-start, skewed, or sparse data; CMF works better in domains with tall user-item matrices, CD-SVD cannot handle the extra noise added through source domain ratings, and CD-CCA assumes that it has access to a full matrix of ratings.

Table 1: Proposed and baseline algorithms and their properties

| Domain | Algorithm | Approach | Sharing | Input data type | Weakness | Speed |
|---|---|---|---|---|---|---|
| Single | SD-SVD | Matrix factorization on target domain | - | continuous | Single domain | Fast |
| Cross | CD-SVD | Matrix factorization<br>Union source-target domains | Shared users | continuous | Not explicit cross-domain, weak noise handling | Relatively fast |
|  | RMGM | Shared user-item clusters<br>Joint user-item mixture and clusters | No sharing | categorical | weak handling of skewness, sparsity, extreme cold-start | Very slow |
|  | CMF | Matrix factorization,<br>Joint domains,<br>Shared user factors | Shared users | continuous | weak handling of fat user-item matrices,<br>noise | Relatively slow |
|  | CD-CCA and CD-LCCA | CCA-based factorization | Shared users | continuous | Assumes full matrices, needs more users vs. items | Fast |

# 4.0 DATASETS

We use the following three datasets for carrying our experiments in this thesis: the Yelp dataset, the Imhonet dataset, and the Supermarket dataset. Each of these datasets have different characteristics that make them suitable for our purposes and provide different views to the proposed analyses. Each of the datasets contain a different nature of items in the domains: the Yelp dataset contains user ratings or preferences on “business services”, the Imhonet dataset contains user ratings or preferences on online items, and the Supermarket dataset includes the actual supermarket purchase history of customers. The Yelp and Supermarket datasets have an average size with more domains, while the Imhonet dataset is a large dataset with four domains. The Yelp and Imhonet datasets include user ratings; while the Supermarket dataset includes the amount of money the customers have spent on goods and their purchase frequency . Table 2 shows these characteristics of each of the datasets. We explain each of these datasets in the following sections.

## 4.1 YELP ACADEMIC DATASET

The Yelp academic dataset is available in http://www.yelp.com/academic_dataset by Yelp for academic purposes. The dataset contains user reviews on business services from various categories and subcategories. It includes a business category hierarchy with 510 unique categories and 21 parent business categories (or super categories). As an example “Active Life” is a parent category with subcategories, such as “Aquariums” and “Diving”, and sub-subcategories, such as “Scuba Diving”. User reviews include a textual review plus user ratings on the service. Each rating can be between one and five stars. Each business

Table 2: General Characteristics of Datasets

| | Yelp | Imhonet | Supermarket |
|---|---|---|---|
| Type of Domain | business services | online items | supermarket goods |
| Type of Feedback | ratings + reviews | ratings | purchase history |
| Number of Domains | 21 | 4 | 22 |
| Data Size | average (>100K) | large (>1M) | average (>100K) |
| Average Sparsity | sparse | most sparse | least sparse |
| Average Skewness | skewed | most skewed | least skewed |
| Shape of User-Item Rating Matrix | tallest | both tall and fat | fattest |
| Sample Domains | Restaurants,<br>Health services,<br>Home services | Movies,<br>Games,<br>Perfumes, Books | Fruits and vegetables,<br>Outdoor appliances |

Table 3: Basic Statistics for Yelp Academic Dataset.

| | Min Review Num | Max Review Num | Average Review Num | Median Review Num | Review Num Variance | Average Review Score | Median Review Score | Review Score Variance |
|---|---|---|---|---|---|---|---|---|
| Businesses | 3 | 862 | 20.19 | 6 | 1874.6 | 3.67 | 3.5 | 0.7437 |
| Users | 0 | 5807 | 38.86 | 7 | 13901.48 | 3.74 | 3.86 | 0.9320 |

can be related to more than one categories and parent categories. For example, a restaurant can belong to both “Food” and “Restaurants” parent categories, while a grocery store only belongs to the “Foods” category. For simplicity, we consider only one parent category for each business. We try to pick the most descriptive category for each business; e.g. “Restaurants” for a restaurant.

In the Yelp academic dataset, there are 229, 908 reviews on 11, 537 businesses from 43, 874 users. The reviews are gathered from local businesses of four states: Arizona, California, South Carolina, and Colorado. Table 3 shows some basic statistics from this dataset.

Preprocessing of this dataset includes the following steps: importing the dataset from JSON format to the sparse matrix format, finding the shared users across category pairs, reducing the domain pairs’ records to include only the shared users and the rated items by them, separating the category pairs with enough information for the analysis, and separation of test, train, and evaluation data.

## 4.2 IMHONET DATASET

Imhonet dataset is an anonymized dataset obtained from an online Russian social system called Imhonet. Imhonet is relatively unique in several aspects including its diverse nature. It allows users to rate and review a range of items from books and movies to mobile phones

Table 4: Basic Statistics for Imhonet Dataset.

| | Book | Game | Movie | Perfume |
|---|---|---|---|---|
| user size | 362448 | 72307 | 426897 | 19717 |
| item size | 167384 | 12768 | 90793 | 3640 |
| density | 2.22E-04 | 0.0014 | 7.30E-04 | 0.0035 |
| record number | 13438520 | 1324945 | 28281946 | 253948 |
| max number of rating per user | 29524 | 1173 | 30014 | 2436 |
| max number of rating per item | 84805 | 9069 | 87848 | 5336 |
| average number of rating per user | 37.0771 | 18.2339 | 6.63E+03 | 12.8796 |
| average number of rating per item | 80.2856 | 103.7708 | 311.4992 | 69.7659 |
| median number of rating per user | 20 | 7 | 20 | 6 |
| median number of rating per item | 3 | 5 | 5 | 7 |
| var of number of rating per user | 1.13E+04 | 1.10E+00 | 3.05E+04 | 894.296 |
| var of number of rating per item | 1.04E+06 | 2.02E+05 | 5.48E+06 | 1.01E+05 |

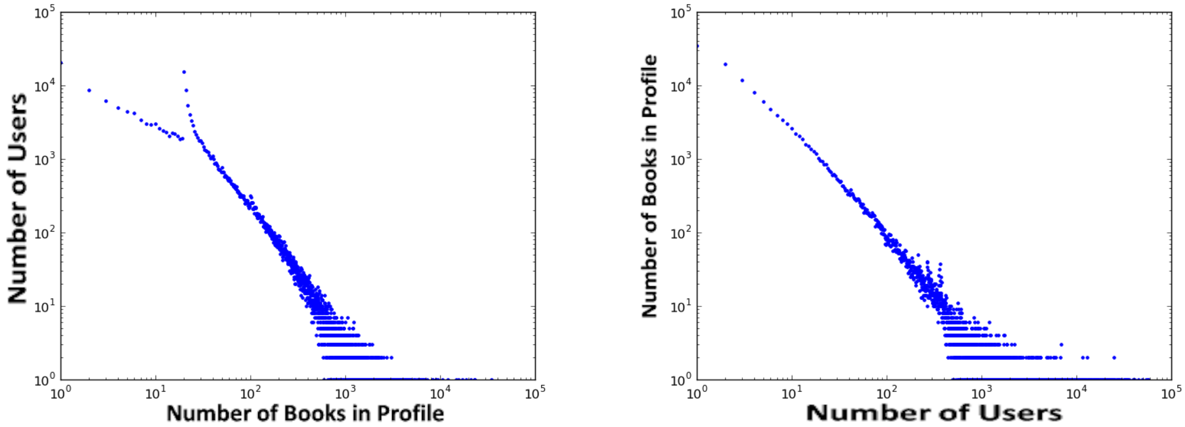
and architectural monuments[1]. This system also contains many aspects of a social network, including friendships, blogs and comments. We use a dataset that includes four sets of ratings - on books, movies, games, and perfumes. Each rating record in the dataset includes a user ID, an item ID, and a rating value between zero (not rated) and ten. The same user ID indicates the same user across the sets of ratings.

Figure. 6 (a) shows the scale of the number of book ratings per user in log-log coordinates and Figure 6 (b) shows the number of ratings for each book. As the figure shows, the plot of number of user raters per book follows the usual power law distribution. But the plot of the number of book ratings per user does not follow a usual pattern. It looks like a combination of two distributions. The same phenomenon happens in the other domains. This peculiar shape is produced by two interfaces for new users that Imhonet offered at different times. One interface asked each new user to rate at least 20 books and movies to receive recommendations. Another interface allowed exploring the system right away adding ratings one by one. To preprocess this dataset we should find the shared users across category pairs, reduce the domain pairs' data to include only the shared users and the rated items by them, separate the category pairs with enough information for the analysis, and separate the test, train, and evaluation data.

### 4.3 SUPERMARKET DATASET

This dataset includes the purchase history of some customers in a large-scale Supermarket in Australia. The data has been gathered during a health study from the Supermarket employees and offered them a 10% discount on fruit and vegetables as part of a health program. The date range is from January 1, 2014 to 31 December, 2014. It is an anonymized dataset of $1,529,055$ records of $1,589$ customers purchasing $35,638$ items. The number of unique user-item purchase records is $736,416$. The number of items in this dataset is much larger than the number of customers and the dataset is very sparse. As a result, the dataset

[1] Recently, Imhonet has limited its domains to movies, TV shows, TV series, games, and books. However, we have access to some of its previous domains' data



(a) Log-log scale of the number of book ratings of users (showing the number of Users having $K$ number of books rated)

(b) Log-log scale of the number of ratings on books (showing the number of books having $K$ number of users rating them)

Figure 6: Distribution of ratings in the book domain

needs to be cleaned to remove the items with too few purchase records. Additionally, there are return transactions in the data that should be removed from it. The items are categorized into 204 fine-grained categories, such as “salad bar”, “sushi”, “baby wear”, “laundry”, and “floral”. The number of records in each category, ranges from one to $111,485$. Since the categories are fine-grained and, in some cases, overlapping, we should redefine the categories manually. To do this, we categorize the data into 22 main domains. The mapping of this categorization is shown in Appendix A. A summary of basic statistics for the Supermarket domains is shown in Table 5. The purchase history data includes the quantity of purchase and the amount of money spent on the purchase. To convert this data into preference data, we use tf-idf (term frequency-inverse document frequency) statistics on the item purchase frequency. More specifically, we first build user vectors in the item space by counting the number of times each item is bought by each customer. Then, we discount these vectors by the total number of times that each item has been bought. Eventually, we normalize the user vectors such that the values of purchases for each customer is between zero and one.

Table 5: Basic Statistics for Supermarket Dataset

| domain name | number of customers | number of items | number of distinct customer-items records | number of records | density |
|---|---|---|---|---|---|
| breads | 1519 | 612 | 25571 | 59388 | 0.0275 |
| alcoholic drinks & cigarrettes | 217 | 306 | 580 | 1690 | 0.0087 |
| beauty | 1476 | 3178 | 33827 | 48371 | 0.0072 |
| canned and pickled | 1448 | 929 | 23920 | 45699 | 0.0177 |
| cooking essentials | 1506 | 1748 | 48094 | 71646 | 0.0182 |
| clothing | 1026 | 3757 | 7518 | 8459 | 0.0019 |
| dairy | 1535 | 1150 | 36190 | 105287 | 0.0205 |
| discounts and coupons | 1223 | 504 | 8479 | 20504 | 0.0137 |
| events | 913 | 505 | 4264 | 4870 | 0.0092 |
| fish, meat, poultry and eggs | 1536 | 2080 | 66848 | 149008 | 0.0209 |
| fruit and vegetables | 1549 | 980 | 95166 | 309040 | 0.0626 |
| gifts | 1122 | 573 | 5308 | 6316 | 0.0082 |
| health | 1456 | 1511 | 20827 | 31877 | 0.0094 |
| home indoor | 1454 | 2129 | 19999 | 25101 | 0.0064 |
| home outdoor | 1103 | 518 | 5596 | 6821 | 0.0097 |
| international food | 1358 | 1185 | 15091 | 24230 | 0.0093 |
| leisure | 1121 | 1330 | 5872 | 11723 | 0.0039 |
| pets | 1051 | 1131 | 12312 | 31085 | 0.0103 |
| prepared meals and snacks | 1563 | 5452 | 140042 | 269454 | 0.0164 |
| soft drinks, tea and coffee | 1534 | 1705 | 37201 | 85771 | 0.0142 |
| sweets | 1550 | 2805 | 84612 | 151827 | 0.0194 |

In summary, preprocessing of this dataset includes cleaning the data, re-defining the categories into domains, aggregating the purchase history in time into unique customer-item purchases, converting the purchase history into tf-idf preference data, finding the shared customers across category pairs, reducing the domain pairs' records to include only the shared customers and the rated items by them, separating the category pairs with enough information for the analysis, and separation of test, train, and evaluation data.

## 4.4 SUMMARY

In summary, we are using three different datasets with various characteristics in this dissertation: the Supermarket purchase dataset, the Yelp academic dataset, and the Imhonet dataset.

As we have seen in Table 2, these datasets have different sizes: Yelp and Supermarket datasets are average in size and Imhonet is a large-scale dataset. The nature of domains in these datasets is different: in Yelp, we see user preferences on a whole business service. Thus, there is no specific item that users rate in Yelp. For example, users rate a restaurant based on various factors in the restaurant. But, they do not rate each of the foods that have been served in the restaurant. The Supermarket dataset does not include any ratings. Customer purchase histories and their frequencies are represented in this dataset. Thus, we do not see any user preference in the form of rating in this dataset. The feedback we have in this dataset is of an implicit format. Imhonet is the most standard recommender systems dataset among the three: It includes user preference ratings on each of the items. In such a dataset, not only users purchase or consume an item, but also decide to express their preference on that item by rating it. Consequently, the ratings in datasets with explicit ratings are usually more skewed: users tend to rate the items they like more. Although Imhonet is a typical recommender system dataset, working with it is more difficult because of its large size and few number of domains.

To have a more global view of the differences among these datasets, we show some of their characteristics in the following figures.

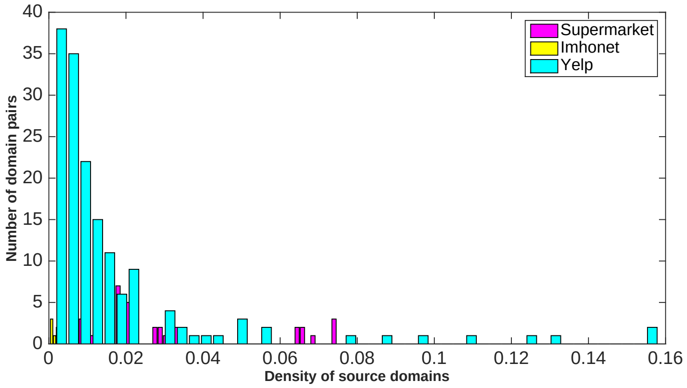


Figure 7: Density of domains in each of the datasets

Figure 7 shows the density of user ratings in different domains in each of the datasets. As we can see in this picture, the Imhonet dataset is very sparse. Also most of the domains have a similar density in this dataset. The Yelp dataset, is also sparse, but less than the Imhonet dataset. In this dataset, most of the domains are very sparse, while there are some domains with much more density. The density of the Supermarket dataset is more than both Yelp and Imhonet datasets. Also, the distribution of densities looks flatter compared to the Yelp dataset.

Figure 8 shows the ratio of number of users to number of items in the domains of each of the datasets. Here, we can see that the number of users compared to number of items can be much larger in the Yelp dataset compared to the other ones. In these cases, the user-item rating matrix, is a tall matrix. However, there are many domains in which the user-item matrix is not very tall in the Yelp dataset. In the Supermarket dataset, we mostly see fat user-item matrices. The ratio of users to items is mostly small in the Supermarket dataset. For the Imhonet dataset, we see both tall and fat user-item matrices for different domains. However, in some of the domains, this ratio is much smaller than the other two datasets.

If we look at the distribution of number of users in the domains, we notice that this distribution is more flat in the Imhonet and Supermarket datasets compared to the Yelp dataset. It means that there are many domains in the Yelp dataset with a few number of

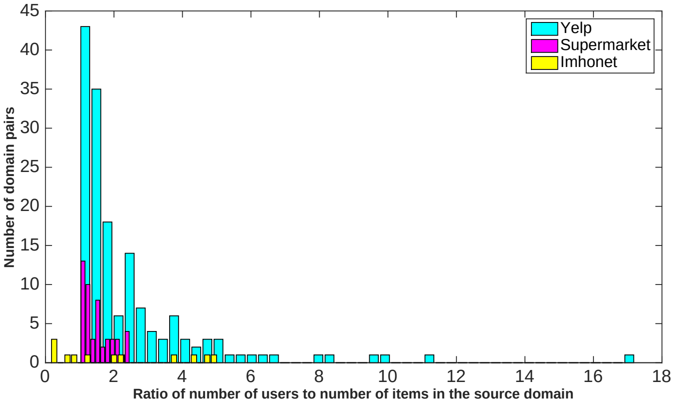


Figure 8: Number of users to number of items ratio in each of the datasets

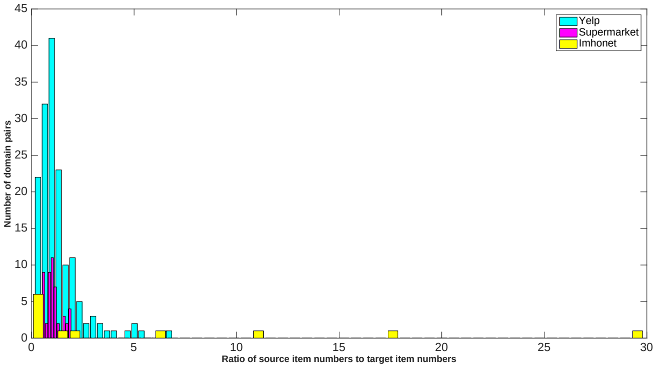


Figure 9: Number of source domain items to number of target domain items in each of the datasets

users and some domains with many users.

Also, looking at Figure 9, we can see that the Imhonet dataset has the maximum ratio of source domain items to target domain items. When this ratio is larger, we have a fatter user-item matrix in the source domain, compared to the target domain. It means that the number of items in domains of Imhonet vary more than the other two datasets. We can see that the Supermarket dataset has the least values for ratio of source domain items to target domain items. As a result, the source domain user-item matrices are not as fat, compared to the target domain user-item matrices.

We expect to see different behaviors in the results for each of these datasets because of their different characteristics. In the following chapters, we analyze how each of the differences can produce different results.

# 5.0 GENERAL EXPERIMENTS: CD-CCA VS. BASELINE ALGORITHMS

The goal of this chapter of thesis is to answer the research question Q1.1. More specifically, we would like to see if the additional data available to cross-domain recommenders help us to provide better recommendations to users; if the cross-domain recommenders can harm the recommendation performance; if there is a cross-domain recommender system that can perform better than other cross-domain recommender systems; and if the improvement we get from the cross-domain recommendations are because of the additional provided data or the properties of the cross-domain algorithm.

To find an answer to the above questions, we use CD-CCA (and CD-LCAA), as one of the cross-domain algorithms, in addition to other state-of-the-art cross-domain and single-domain algorithms that are mentioned in Section 3.3. We compare the performance of these algorithms using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) of the recommended items. To understand the effect of having additional data on the recommendation performance, we apply the single-domain algorithm only on the target domain data and the cross-domain algorithms on both source (auxiliary) and target datasets and compare their results. Additionally, to understand the effect of approach on the recommendation results, we apply the single-domain algorithm on a combination of source and target data to have a fair comparison with cross-domain algorithms. This setting is shown in Figure 10. In the next step, we examine the correlation between these algorithms' performances, on the available domain-pairs in the data, to understand if an increase in the performance of each of the algorithms can lead us to an increase in other algorithms' performance.

Our hypothesis in this part of analyses is:

- It is possible that adding additional domains' data harm the recommendation results;

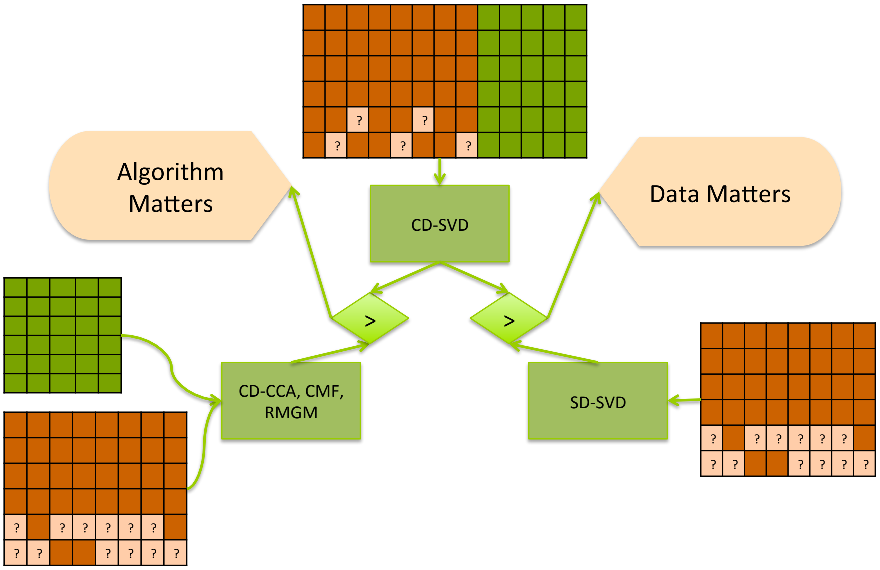


Figure 10: Experiment setup to answer research question Q1.1

- However, using auxiliary material, if selected and applied correctly, should either improve or preserve the performance of recommender systems;
- The performance of single-domain and cross-domain recommenders are correlated with each other due to the data characteristics;
- However, the improvement achieved by using auxiliary data depends also on the applied algorithm.

In the following sections, we present the results of our proposed and baseline algorithms on each of the datasets.

## 5.1 EXPERIMENT SETUP

To run the experiments on each of the datasets, we implement a user-stratified 5-fold cross-validation setting. The user-stratified setting is used to represent a common situation that happens with recommender systems: we would like to predict the ratings of some (probably

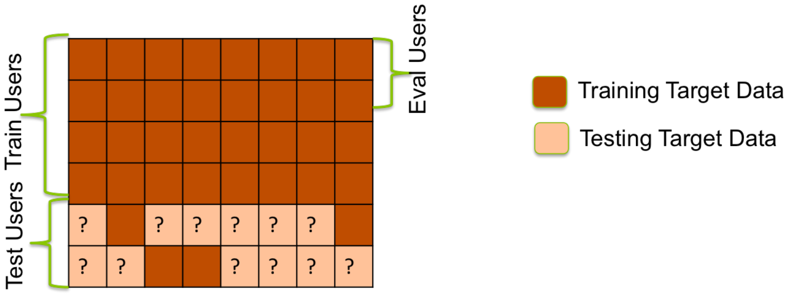


Figure 11: Separating test, train, and evaluation data from the target domain

new) users, given that we have the ratings of other (probably similar) users. As a result, some of the users (20%) are selected as test users and the rest of them (80%) are selected as training users. 80% of the ratings for the test users on the items in target domain is removed randomly from the training dataset. The algorithms approximate this 80% of test user ratings based on the training set. Eventually, the approximated test ratings are compared to the real ones to calculate the error of algorithms.

The reason to remove 80% of test user ratings, and not all of their ratings, is to avoid the extreme cold-start case and to be able to perform a cold-start analysis on the user profile sizes. Thus, we use a random 20% selection of each test user's rating and estimate the rest of test users' ratings (the removed 80%) conditioned on observing this 20% of their ratings and the ratings of users in the training set. Having this setting, if a test user has a large profile in the target domain, we will have more information on this user, compared to another test user with a small target domain profile. Consequently, the distribution of profile sizes among the test users is a factor of the gold-start profile sizes distribution. Thus, the amount of information that we have from the test users is kept in accordance with the amount of information we have from them as the gold-standard. This allows us to perform a cold-start analysis that is similar to the real-world setting: some new users are active and have more ratings in the beginning of using a system, while others have less ratings.

Some of the algorithms have parameters that should be selected by cross-validation. For example, the number of components should be provided as an input to the SVD++

algorithms. To find the best set of parameters for each algorithm, we remove a "validation" set of ratings from the training data. Selection of this validation set is in accordance with selection of the test set; we select 15% of users as validation users and remove 80% of their ratings from the training set. Then, we train the algorithms with different values of parameters on the remaining training ratings and test it over the validation dataset to select the parameters that result in the best performance.

After selecting the best parameters, we add the validation set data to the training set; train the algorithms based on this new training dataset; and test it on the test data of the removed 20% of users. Figure 11 shows a toy example of separating the test, train, and evaluation data in a target domain.

We repeat these experiments 5 times, each time selecting a different set of test users, for the 5-fold cross-validation. Eventually, we average over the performance of algorithms in these 5 times and report it.

For the single-domain algorithm, we use only the target domain dataset. However, for cross-domain algorithms, we have both source and target datasets. To be able to compare single and cross-domain algorithms, we remove the same set of ratings for all of the algorithms. Thus, for each test user in the cross-domain algorithms, we have all of the users' ratings from the source domain, plus 20% of her ratings in the target domain, as training data. The remaining 80% of test user's target domain ratings is what we test the algorithms on.

To measure the performance of algorithms, we use Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Although there are other performance measures, such as ranked-based measures like nDCG, precision, and recall, that can be used in the recommender systems field, we choose RMSE and MAE because of the way we formalize our problem. The proposed algorithms are formulated as estimating user ratings over the items. Consequently, the closeness of the estimated rating to the real rating is the measure that is important to us. If $R$ is the set of test ratings, $r_{u,i}$ is the rating of user $u$ on item $i$, and $\hat{r}$ is the estimated rating by the algorithm, then RMSE and MAE can be calculated as in Equations 5.1 and

5.2.

$$\text{RMSE} = \sqrt{\frac{\Sigma_R (r_{u,i} - \hat{r}_{u,i})^2}{|R|}} \tag{5.1}$$

$$\text{MAE} = \frac{\Sigma_R \, \text{abs}(r_{u,i} - \hat{r}_{u,i})}{|R|} \tag{5.2}$$

Since we have normalized vectors of purchase frequencies in the Supermarket dataset (instead of user ratings), we cannot use the RMGM algorithm directly on this data. The frequency rates in the Supermarket dataset are normalized and have a value between zero and one. To run RMGM on this dataset, we convert these frequencies to a 10-scale categorical values. To do this, we multiply each of the normalized frequency values by ten and use the ceiling value of it.

## 5.2 RESULTS OF THE SUPERMARKET PURCHASE DATASET

In this section we compare the result of the proposed cross-domain algorithm (CD-CCA) with the baseline cross-domain and single-domain algorithms. As mentioned in Section 4.3, we use the 22 parent categories as domains for cross-domain recommendation. Considering each domain once as the source domain and once as the target domain, we end up with 462 domain pairs. However, we run the experiments only on the domain pairs that have larger number of users compared to items (in both source and target domains). Consequently, we end up with 50 domain pairs in this dataset. These 50 domain pairs, and some basic statistics are presented in Tables 37 to 39 in Appendix A.2.

As explained in Section 5.1, we use user-stratified 5-fold cross-validation to run each of the algorithms on each of the domain pairs. The results of running algorithms on domain pairs is presented in Table 40 of Appendix A.3.

To have a better global view of these results, we plot the RMSE of these five algorithms on each of the domains in Figure 12 and the MAE of these algorithms in Figure 13. The $X$-axis shows each of the domain pairs and the $Y$-axis shows the error reported by the algorithms. The domain pairs are ordered based on CD-CCA's error on them. The reported errorbars are for p-value $< 0.05$ based on the 5-fold cross-validation. As we can see in these pictures, RMGM (shown with yellow star marker) and CD-CCA (shown with red circle marker) perform significantly better than other algorithms, in most of the domain pairs. RMGM performs by far better than other algorithms in most of the domain-pairs. While these two cross-domain algorithms perform well, CD-SVD (shown with blue square markers) and CMF (shown with purple cross markers) algorithms have a high error rate in most of the domain pairs. In many cases, the single-domain SD-SVD algorithm (shown with green diamond markers) performs better than these two cross-domain algorithms.

To be more exact, we can look at the number of domain pairs in which each of the two algorithms have a significant difference in their reported error. Table 6 shows this relationship between RMSE of algorithms and Table 7 shows it between MAE of them. The table cell related to $i^{th}$ row and $j^{th}$ column shows the number of domain pairs in which the $i^{th}$ algorithm performed significantly better than the $j^{th}$ algorithm. The last column (row) of the table shows the total number of domain pairs in which the algorithms performed significantly better (worse) than other algorithms. Thus, the higher column-sum and the lower row-sum of an algorithm indicates a generally better performing algorithm in all of the domain-pairs. As we can see in these tables, RMGM is better than other algorithms in 149 comparisons on domain pairs, measured by RMSE, and 159 comparisons, measured by MAE. In 32 and 30 domain pairs RMGM performs significantly worse than other domain pairs measured by RMSE and MAE, respectively. CD-CCA is the next-best algorithm, performing significantly better than other algorithms in 129 and 109 comparisons on RMSE and MAE of domain pairs. CMF is the worst algorithm on this dataset and CD-SVD is the second worst algorithm in terms of the number of domain pairs with significantly higher RMSE and MAEs.

As an interesting observation based on Tables 6 and 7, none of the cross-domain algorithms are always better than the single-domain algorithm. Although the number of domain pairs in which the cross-domain algorithms perform better than SD-SVD varies, there are always some domain-pairs in which SD-SVD performs better than the cross-domain algorithms. Even RMGM, which is the best-performing cross-domain algorithm in this dataset,

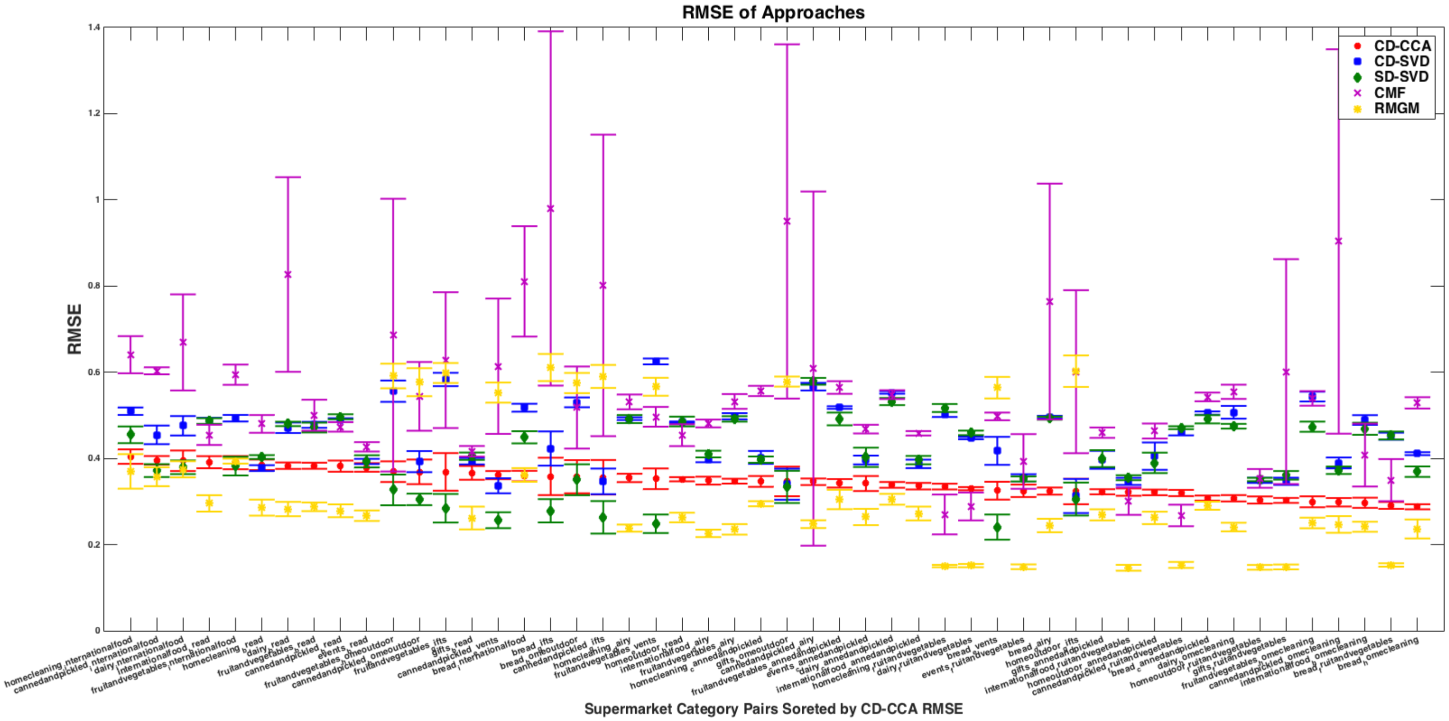


Figure 12: RMSE of algorithms on 50 Supermarket domain pairs ordered by the RMSE of the CD-CCA

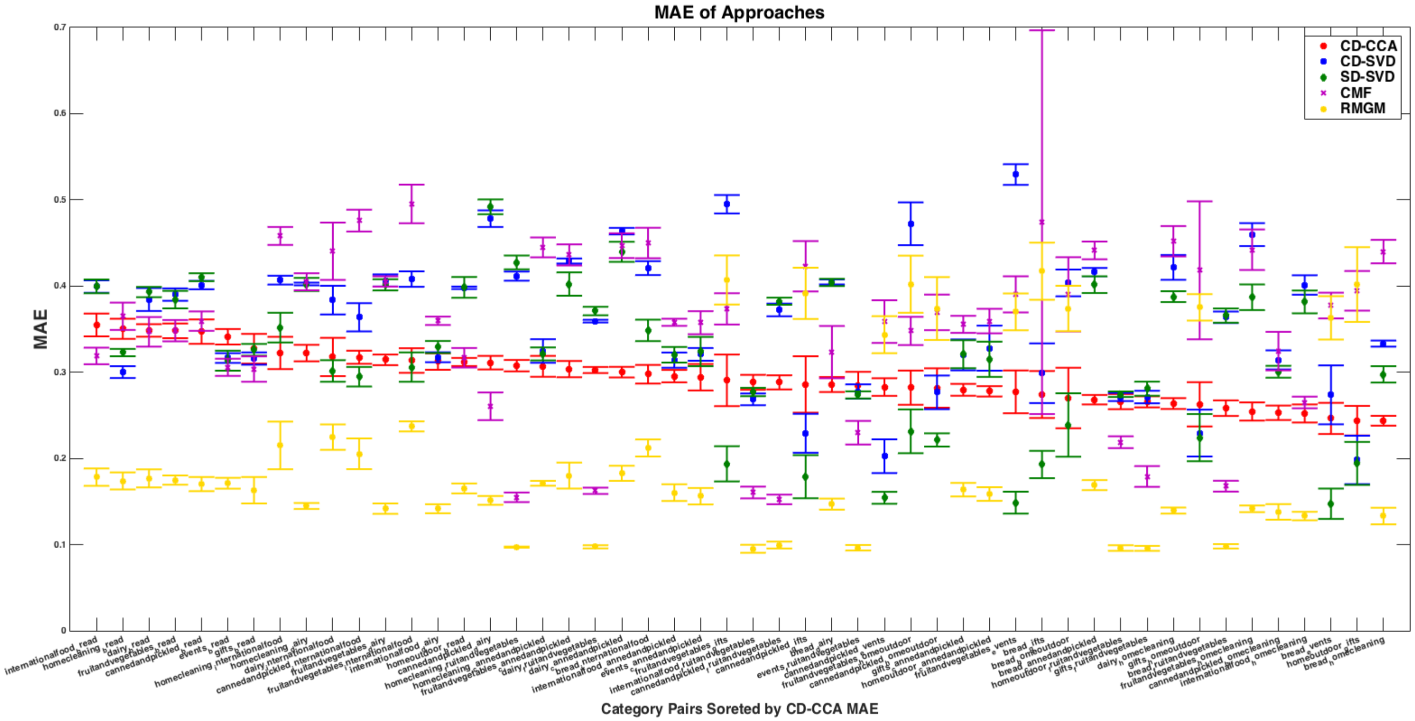


Figure 13: MAE of algorithms on 50 Supermarket domain pairs ordered by the MAE of the CD-CCA

Table 6: Number of domain pairs with significant RMSE difference among algorithms. Each row shows the number of domain pairs in which the algorithm of that row works significantly better than the algorithms mentioned in the columns.

| RMSE | CD-CCA | CD-SVD | SD-SVD | CMF | RMGM | SUM |
|---|---|---|---|---|---|---|
| CD-CCA | 0 | 42 | 35 | 41 | 11 | 129 |
| CD-SVD | 0 | 0 | 4 | 27 | 8 | 39 |
| SD-SVD | 7 | 20 | 0 | 31 | 11 | 69 |
| CMF | 3 | 9 | 7 | 0 | 2 | 21 |
| RMGM | 35 | 40 | 36 | 38 | 0 | 149 |
| SUM | 45 | 111 | 82 | 137 | 32 | |

Table 7: Number of domain pairs with significant MAE difference among algorithms. Each row shows the number of domain pairs in which the algorithm of that row works significantly better than the algorithms mentioned in the columns.

| MAE | CD-CCA | CD-SVD | SD-SVD | CMF | RMGM | SUM |
|---|---|---|---|---|---|---|
| CD-CCA | 0 | 34 | 27 | 30 | 11 | 102 |
| CD-SVD | 6 | 0 | 3 | 19 | 7 | 35 |
| SD-SVD | 12 | 19 | 0 | 28 | 11 | 70 |
| CMF | 11 | 19 | 17 | 0 | 1 | 48 |
| RMGM | 39 | 42 | 39 | 39 | 0 | 159 |
| SUM | 68 | 114 | 86 | 116 | 30 | |

performs significantly worse than SD-SVD in 11 domain pairs out of the 50 domain pairs.

Another interesting observation is that there is no absolute best algorithm for this dataset. Looking at both of these tables, we can see that there is no row or column with all-zero values. This means that there is no algorithm that always performs significantly better than (or similar to) other algorithms. The only case in which one of the algorithms always performs significantly better or similar to another one is CD-CCA compared to CD-SVD based on RMSE. In none of the domain pairs, CD-SVD performs significantly better than CD-CCA.

Tables 41 and 42 in Appendix A.3 show a detailed view of comparison of algorithms in each of the domains. Each column of these tables ("Alg A > Alg B") shows in which domains the left-hand side algorithm ("Alg A") is performing better than the right-hand side one ("Alg B"). As we can see, there are $1000 = 50 \times 20$ cells in each of the tables, representing the combination of domain pairs with each two of the algorithms. Out of these 1000 combinations, the two tables are different in 103 cells. This means that, in the 897 remaining experiments, the algorithms performed similarly compared to each other, given either RMSE or MAE error measures. Also, we do not see any algorithm that performs better than any other algorithm in all of the domain pairs (otherwise, we would have had the value 50 in the reported tables).

As we can see in these tables, there are 16 domain pairs in which all of the cross-domain algorithms perform either significantly better than, or similar to, SD-SVD based on the RMSE measure. Looking at the MAE measure, there are 20 domain pairs in which adding the source domain information, using all of the cross-domain algorithms, either increases the performance significantly, or does not change it. As example of these domain pairs, we can name "bread →[1] dairy", "dairy → fruit and vegetables", and "international food → home cleaning". There are four domain pairs in which the RMSE and MAE measures are not agreeing on, in terms of having all cross-domain algorithms performing better than SD-SVD. An example of these domain pairs is "canned and pickled → home cleaning". While CMF works similar to SD-SVD based on MAE measure in this domain pair, the RMSE of

[1] The arrow shows the direction of transferring information from the source domain to the target domain. The left-hand side domain is the source domain; and the right-hand side one is the target domain.

SD-SVD is significantly better than CMF.

Also, there are 6 and 8 domain pairs in which SD-SVD performs significantly better than all of the cross-domain algorithms based on the RMSE and MAE measures respectively. “bread → events”, “canned and pickled → gifts”, and “fruit and vegetables → gifts” are examples of these domain pairs. The disagreement between RMSE and MAE measures, in SD-SVD performing better than all of the cross-domain algorithms, comes from the “bread → gifts” and “fruit and vegetables → home outdoor” domain pairs. While SD-SVD performs significantly better than CD-CCA based on RMSE in “fruit and vegetables → home outdoor”, they do not have a significant difference using MAE measure. For the “bread → gifts” domain pair, the disagreement is on comparing SD-SVD and CMF algorithms.

Although there are significant differences between the performances of algorithms, in many cases their errors on domain pairs are correlated with each other. Table 8 shows the correlations between the RMSE of these algorithms. The numbers with star represent a significant correlation with p-value $< 0.01$. Figure 14 shows the scatter plot of RMSE of these algorithms of all of the domain pairs. The general observation with the correlation between RMSE of algorithms in this dataset suggests that the correlation among cross-domain RMSEs are either significantly positive or non-significant. However, the correlations between SD-SVD and cross-domain algorithms is mostly negative. The exception is the correlation between RMSE of SD-SVD and CD-SVD. Despite other cross-domain algorithms, CD-SVD’s RMSE is higher when SD-SVD works worse; and vice versa. A similar pattern can be seen in the correlation of MAE of these algorithms on the Supermarket dataset. Figure 15 shows the scatter plot of MAE of algorithms in all of the domains. We can see that the cross-domain algorithms are positively correlated with each other.

## 5.3 RESULTS OF THE YELP DATASET

In the Yelp dataset, we have 21 parent categories. We use the star ratings of reviews within each category. For each pair of categories, we find out the common users (the users who have rating reviews in both of the selected domains). To obtain more reliable results, we exclude

Table 8: Correlation between RMSE of algorithms on all domain pairs in the Supermarket dataset

| RMSE Correlation | CD-CCA | CD-SVD | SD-SVD | RMGM | CMF |
|---|---|---|---|---|---|
| CD-CCA | 1 | 0.199 | -0.0537 | 0.39∗ | 0.1702 |
| CD-SVD | 0.199 | 1 | 0.4342∗ | 0.0679 | -0.045 |
| SD-SVD | -0.0537 | 0.4342∗ | 1 | -0.6734∗ | -0.2906 |
| RMGM | 0.39∗ | 0.0679 | -0.6734∗ | 1 | 0.5049∗ |
| CMF | 0.1702 | -0.045 | -0.2906 | 0.5049∗ | 1 |

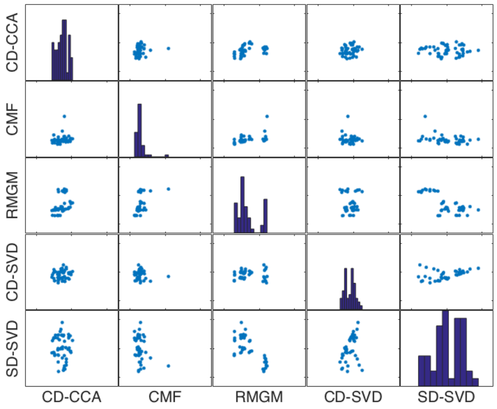


Figure 14: Scatter plot of RMSE of algorithms on 50 Supermarket domain pairs

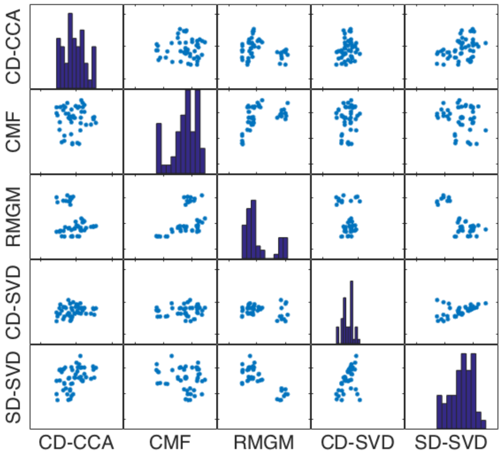


Figure 15: Scatter plot of MAE of algorithms on 50 Supermarket domain pairs

the category pairs, within which the number of common users is smaller than the number of items in any of the two categories. For each pair of categories, we run the experiments twice: once with the first category as the source and the second category as the second domain, and once the other way around. Eventually, we end up with 158 category (domain) pairs. A summary of these data statistics is shown in Table 9.

Table 9: Summary of domain pair statistics in Yelp dataset

| | Min | Max | Mean | Median |
|---|---|---|---|---|
| User Size | 9 | 11013 | 1064.09 | 424 |
| Item Size | 8 | 4435 | 406.89 | 252.5 |
| Rating Density | 0.0017 | 0.1581 | 0.017 | 0.0084 |

We run CD-CCA, CD-SVD, CMF, and RMGM cross-domain algorithms and SD-SVD single-domain algorithm on the 158 domain pairs in the data. We evaluate the algorithms based on RMSE and MAE. Figure 16 shows the RMSE of all three algorithms on the 158 domain pairs, including the 95% confidence interval; and Figure 17 shows the MAE of the algorithms on these domain pairs. To better comprehend the difference between algorithms, we order the domain pairs based on RMSE and MAE of CD-CCA algorithm on them. Due to the visualization limitations, we cannot show the name of all domain pairs in the picture.

However, it can be seen that in most of the domain pairs, CD-CCA has a lower RMSE compared to both cross-domain and single-domain algorithms. More specifically, CD-CCA always performs significantly better or similar to SD-SVD and CD-SVD, in terms of RMSE. Compared to RMGM and CMF, CD-CCA has a significantly lower RMSE in 112 and 46 of the domain pairs, respectively. The number of domains in which there is a significant difference (0.05 p-value) between RMSE of algorithms are listed in Table 10. Each cell of the table shows the number of domain-pairs in which the algorithm mentioned in its row performs better that the algorithm mentioned in its column. As we can see in the table, CD-CCA and CD-SVD always perform significantly better than the single-domain algorithm (SD-SVD) in terms of RMSE. However, the RMSE in CMF and RMGM is sometimes (in 18 and 83 domain pairs) significantly higher than the single-domain algorithm. Based on the sum of the number of significant differences in domain-pair RMSEs, we can see that CD-CCA is performing better than all of the baseline algorithms: it has the most sum of significantly better RMSE on domain-pairs, and least sum of significantly worse RMSE. CMF is the next best algorithm based on this measure. However, looking at Figure 16, we can see that CMF has a fluctuating and non-steady behavior, compared to other algorithms. In some of the domain-pairs, CMF performs much better than all other algorithms. While in others, it works much worse than the rest. In other words, when there is a significant difference between RMSE of CMF and other algorithms, this difference is mostly large. Additionally, the errorbars for the results of CMF, when it performs poorly, are very wide. This shows that the results of CMF are not as reliable in these domain pairs. Also, we can see that although RMGM performs significantly better than CD-CCA and CMF in 4 and 11 domain-pairs, it is the worst algorithm in terms of the sum of number of significant relationships between RMSE of algorithms.

Table 47 in Appendix B.2 shows the significant comparison details of algorithms, based on RMSE, in each of the domain pairs. Based on this table, there are 67 domain pairs in which all of the cross-domain algorithms are working significantly better than the single-

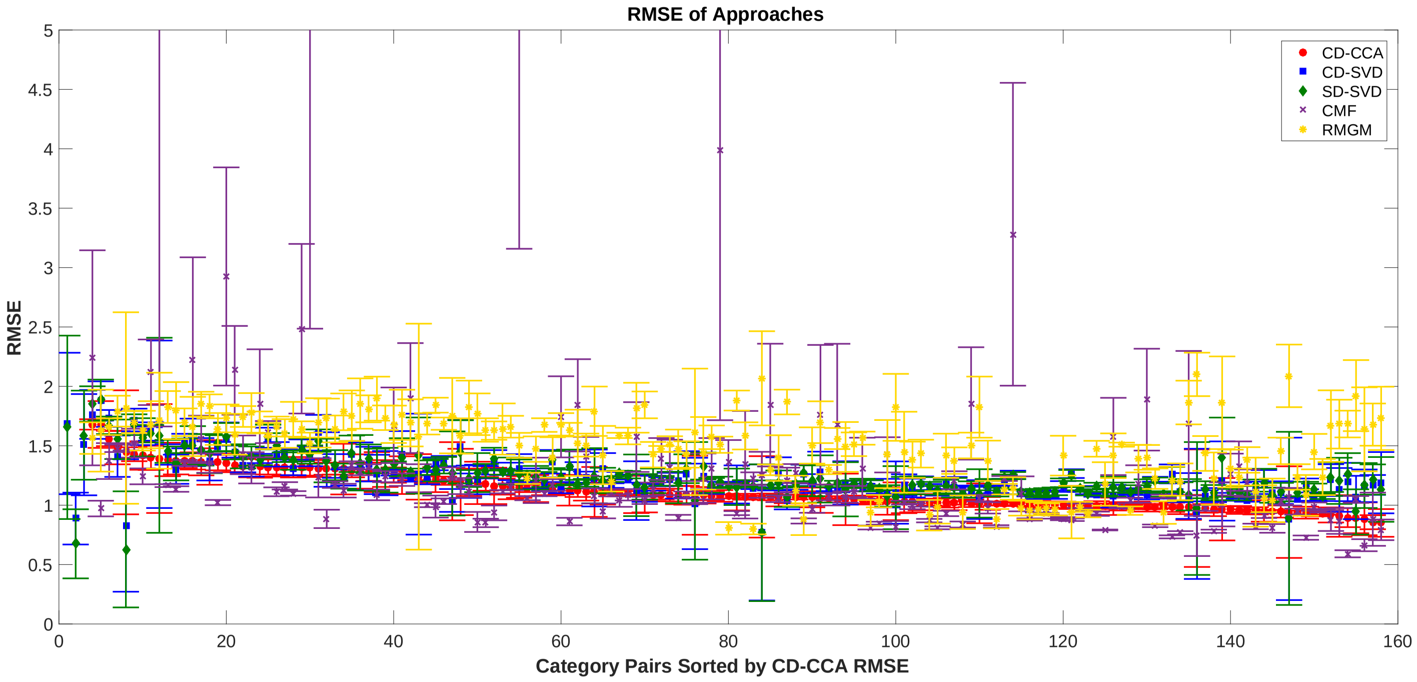


Figure 16: RMSE of algorithms on 158 Yelp domain pairs ordered by the RMSE of the CD-CCA

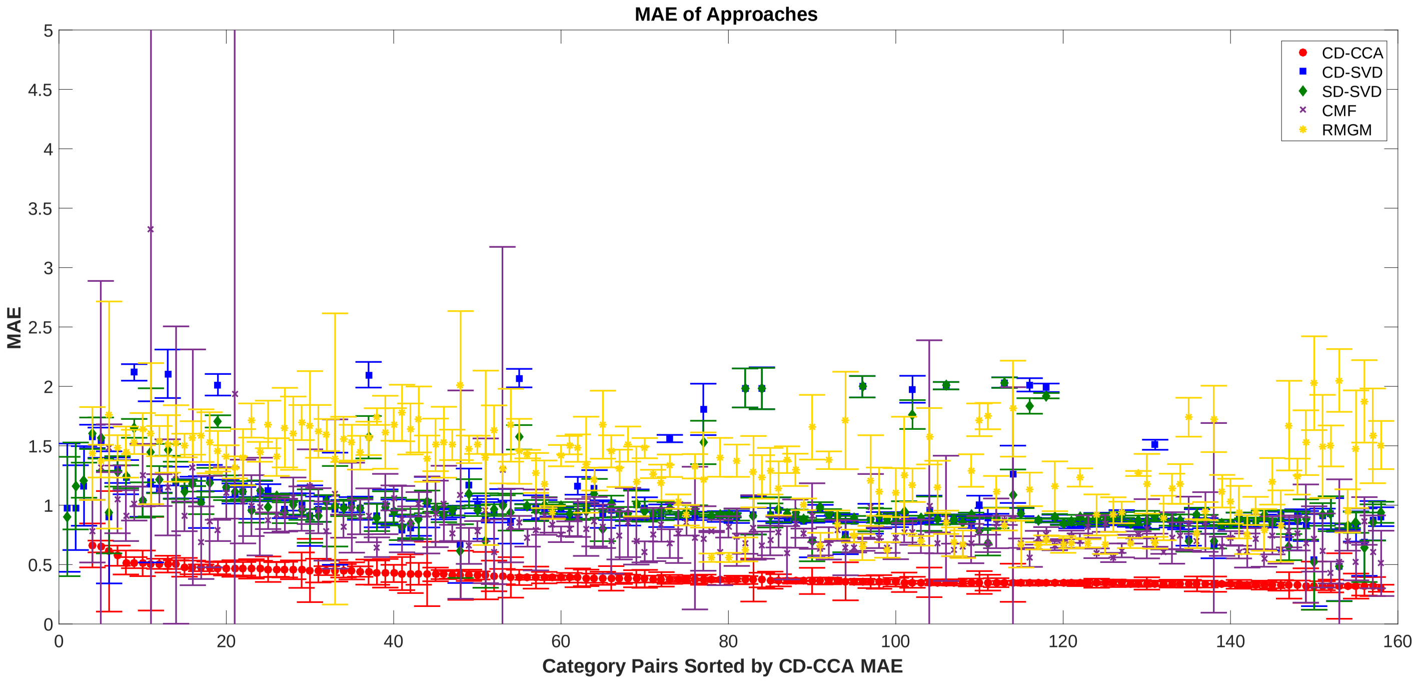


Figure 17: MAE of algorithms on 158 Yelp domain pairs ordered by the MAE of the CD-CCA

Table 10: Number of domain pairs with significant RMSE difference among algorithms for the Yelp dataset. Each row shows the number of domain pairs in which the algorithm of that row works significantly better than the algorithms mentioned in the columns.

| Significant RMSE difference | CD-CCA | CD-SVD | SD-SVD | CMF | RMGM | SUM |
|---|---|---|---|---|---|---|
| CD-CCA | 0 | 74 | 77 | 33 | 112 | 296 |
| CD-SVD | 0 | 0 | 9 | 22 | 87 | 118 |
| SD-SVD | 0 | 0 | 0 | 18 | 83 | 101 |
| CMF | 46 | 75 | 73 | 0 | 91 | 285 |
| RMGM | 4 | 18 | 19 | 11 | 0 | 52 |
| SUM | 50 | 167 | 178 | 84 | 373 |  |

domain algorithm. As examples of these domain pairs, we can name “Active Life → Home Services”, “Beauty & Spas → Arts & Entertainment”, and “Hotels & Travel → Nightlife”. Since CD-CCA always works significantly better, or similar, to SD-SVD, there is no domain pair in which the single-domain algorithm performs better than all of the cross-domain ones.

Although CD-CCA is generally having a lower RMSE compared to the baselines, there are only 13 domain pairs in which CD-CCA performs significantly better than all of the baseline algorithms. These domain pairs include “Nightlife → Food”, “Active Life → Arts & Entertainment”, and “Arts & Entertainment → Event Planning & Services”.

Looking at the MAE of these algorithms in Figure 17, we can see that CD-CCA is performing by far better than all other algorithms in all domain pairs. CMF is the next best algorithm in terms of MAE and RMGM is the worse one. Looking at the details of number of domain pairs with a significantly different MAE for each two algorithms in Table 11, we can see that CMF and CD-CCA cross-domain algorithms always perform significantly better than, or similar to, the SD-SVD single-domain algorithm. However, in 14 and 86 domain pairs, the single-domain algorithm performs better than RMGM and CD-SVD. In the MAE results, we see less fluctuation for the CMF algorithm, compared to the RMSE results. However, we can still see the wide errorbars in some of the domain pairs.

Looking at Table 47 in Appendix B.2, we can see that in 63 of the domain pairs, MAE of all cross-domain algorithms is better than MAE of the single-domain algorithm. These domain pairs include "Public Services & Government → Mass Media", "Pets → Event Planning & Services", and "Health & Medical → Active Life". This number is less than the number of domain pairs with the RMSE-based comparison of all cross-domain algorithms versus the single-domain one. This disagreement comes from domain pairs such as "Active Life → Arts & Entertainment", "Arts & Entertainment → Nightlife", "Automotive → Event Planning & Services", "Beauty & Spas → Active Life", and "Education → Arts & Entertainment". In some of these domain pairs all cross-domain algorithms are significantly better than SD-SVD, measured by RMSE (MAE), while not all of them are significantly better than SD-SVD measured by MAE (RMSE).

In contrast to the low number of domain pairs in which CD-CCA had a significantly better RMSE compared to all other baselines, CD-CCA is having a significantly less MAE in 119 domain pairs. Some of the domain pairs in which CD-CCA performs better measured by MAE, compared to RMSE, are "Active Life → Automotive", "Arts & Entertainment → Beauty & Spas", and "Nightlife → Shopping".

The difference between the MAE and RMSE results can be because of their emphasis on different types of errors. While in the MAE measure the error calculated on all of the datapoints are weighted equally, the RMSE measure puts more weight on the larger errors per datapoint. Based on the above-mentioned results, CD-CCA performs better when using the MAE measure compared to the RMSE measure. On the other hand, CMF works generally better when measured by RMSE compared to MAE. This can mean that there are less large errors happening in the CMF algorithm compared to CD-CCA; However, the total of error made by CD-CCA is smaller than CMF.

We calculate the correlation between error rates of all algorithms in all of the domain pairs (Table 12 and Figure 18). Based on these results, the RMSE of algorithms (except for CMF and RMGM) are significantly correlated. Most notably, the RMSE of SD-SVD and CD-SVD are highly correlated. This correlation is smaller between CD-SVD and SD-SVD with CD-CCA, RMGM, and CMF. We can conclude that if the RMSE of single-domain recommender

Table 11: Number of domain pairs with significant MAE difference among algorithms for the Yelp dataset. Each row shows the number of domain pairs in which the algorithm of that row works significantly better than the algorithms mentioned in the columns.

| MAE | CD-CCA | CD-SVD | SD-SVD | CMF | RMGM | SUM |
|---|---|---|---|---|---|---|
| CD-CCA | 0 | 148 | 148 | 122 | 153 | 571 |
| CD-SVD | 0 | 0 | 6 | 0 | 86 | 92 |
| SD-SVD | 0 | 14 | 0 | 0 | 86 | 100 |
| CMF | 0 | 67 | 25 | 0 | 91 | 183 |
| RMGM | 0 | 30 | 25 | 4 | 0 | 59 |
| SUM | 0 | 259 | 204 | 126 | 416 | |

Table 12: Correlation of RMSE of algorithms with each other. *: significant with p-value < 0.01; **: significant with p-value < 0.001; ***: significant with p-value < 0.0001

| ***: p <0.0001;<br>**: p <0.001; *:p<0.01 | CD-CCA | CD-SVD | SD-SVD | RMGM | CMF |
|---|---|---|---|---|---|
| CD-CCA | 1 | 0.7896*** | 0.7779*** | 0.4384*** | 0.2925*** |
| CD-SVD | 0.7896*** | 1 | 0.955*** | 0.2484* | 0.285*** |
| SD-SVD | 0.7779*** | 0.955*** | 1 | 0.2729*** | 0.2536** |
| RMGM | 0.4384*** | 0.2484* | 0.2484*** | 1 | 0.1217 |
| CMF | 0.2925*** | 0.285 | 0.2536*** | 0.1217 | 1 |

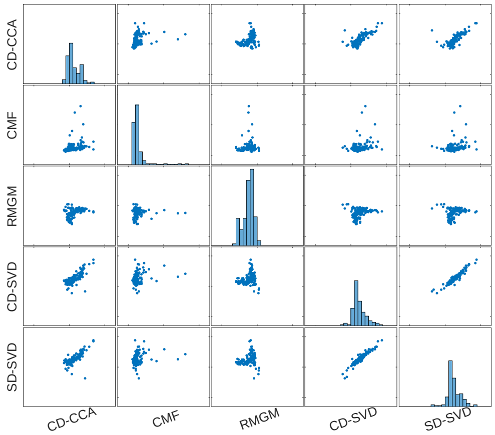


Figure 18: Scatter plot of RMSE of algorithms on 158 Yelp domain pairs

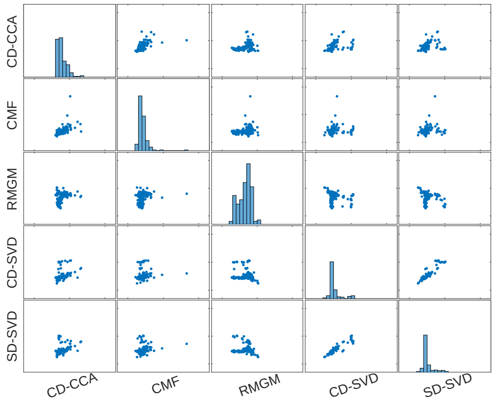


Figure 19: Scatter plot of MAE of algorithms on 158 Yelp domain pairs

is low in the target domain, it is also most likely low for cross-domain recommenders, and vice versa. There is a similar correlation between MAE of these algorithms (Figure 19).

## 5.4 RESULTS OF THE IMHONET DATASET

We have four domains in the Imhonet dataset: books, movies, perfumes, and games. Users can use a rating scale to rate the items in each of these domains from 0 to 10. For ease of comparison among the algorithms and datasets, we normalize ratings by dividing them by the maximum possible rating (10), so that all of them are between zero and one.

Having only four domains, we can have only 12 domain pairs to study on. Since this is a small number of domain pairs compared to the other two datasets, we do not exclude any domain pairs from our analysis of Imhonet dataset. This means that in these domain pairs, the number of users can be smaller than the number of available source or target items. However, for each of the domain pairs, we select the users that have at least one rating in each of the domains and run the experiments on that set of users. Some of the statistics of domain pairs in the Imhonet dataset are presented in Tables 48 to 51 in Appendix C.1. As we can see, in none of the domain pairs, the number of users are more than both source and target domain items.

Based on this table, the Imhonet dataset is much larger than the other two datasets that we are using in this thesis. Especially, the movies and books domains include many users and items. On the other hand, CD-CCA algorithm requires a large memory for loading the data matrices to compute the canonical correlation between the domains. Consequently, we cannot use the regular CD-CCA algorithm for this dataset and we use CD-LCCA instead of it. For the same reason, it is very difficult to run RMGM and CMF algorithms on this dataset, especially considering the running time of these algorithms. Also, running each of these algorithms require a sparse implementation of them. For these reasons, we omit running CMF and RMGM on Imhonet and only compare the results for CD-LCCA, CD-SVD, and SD-SVD. However, we use the name CD-CCA instead of CD-LCCA in the following sections for simplicity.

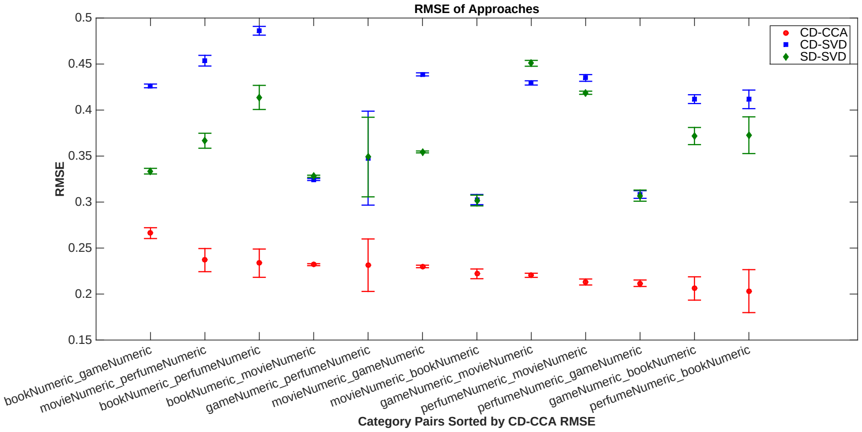


Figure 20: RMSE of algorithms on 12 Imhonet domain pairs ordered by the RMSE of the CD-CCA

Figures 20 and 21 show the RMSE and MAE of algorithms on the 12 domain pairs of Imhonet, sorted by the error of CD-CCA. As explained in previous sections, the reported errorbars represent a 95% confidence interval for the errors.

As we can see in these figures, the single-domain algorithm performs better than, or similar to, CD-SVD in many domains. Only in “book → movie” and “game → movie” domain pairs, we see that CD-SVD is significantly better than SD-SVD. However, CD-CCA performs significantly better than both CD-SVD and SD-SVD in all of the domain pairs.

Also, we can see that in most of the domain pairs the confidence intervals are small. Except for “game → perfume” and “perfume → book” domain pairs, the confidence interval for domain pairs are small. Table 13 shows the number of domain pairs in which each of the algorithms (in rows of the table) is working significantly better than other algorithms (in columns of the table). Note that both RMSE and MAE of algorithms in the Imhonet dataset has the same relationship that is represented in this table. So, in general CD-CCA is the best-performing algorithm in this dataset and SD-SVD is the next best one.

As we have seen in the previous sections, the error of these algorithms are correlated with each other. Figures 22 and 23 show the scatter plots of RMSE and MAE of algorithms in the 12 domain pairs.

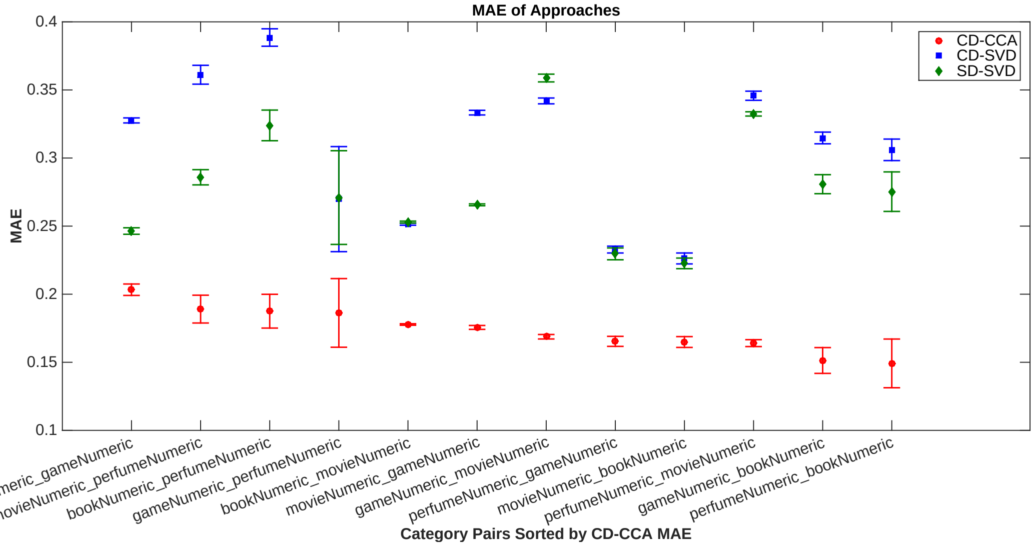


Figure 21: MAE of algorithms on 12 Imhonet domain pairs ordered by the MAE of the CD-CCA

Table 13: Number of domain pairs with significant error difference among algorithms for the Imhonet dataset. Each row shows the number of domain pairs in which the algorithm of that row works significantly better than the algorithms mentioned in the columns.

| | CD-CCA | CD-SVD | SD-SVD | SUM |
|---|---|---|---|---|
| CD-CCA | 0 | 12 | 12 | 24 |
| CD-SVD | 0 | 0 | 2 | 2 |
| SD-SVD | 0 | 7 | 0 | 7 |
| SUM | 0 | 19 | 14 | |

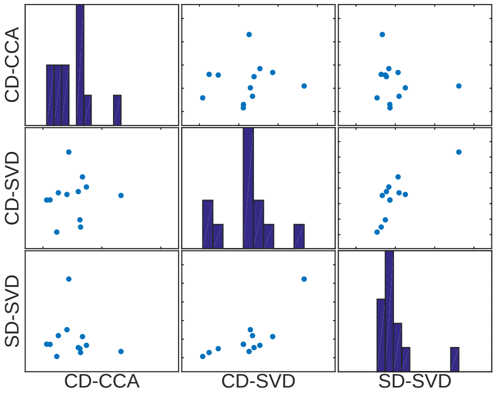


Figure 22: Scatter plot of RMSE of algorithms on 12 Imhonet domain pairs

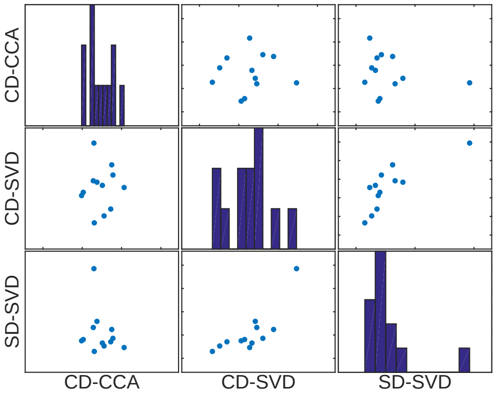


Figure 23: Scatter plot of MAE of algorithms on 12 Imhonet domain pairs

Table 14: Correlation of RMSE of algorithms with each other in the Imhonet dataset. *: significant with p-value < 0.01.

| | CD-CCA | CD-SVD | SD-SVD |
|---|---|---|---|
| CD-CCA | 1 | 0.1993 | -0.1909 |
| CD-SVD | 0.1993 | 1 | 0.7416* |
| SD-SVD | -0.1909 | 0.7416* | 1 |

Table 14 shows the calculated correlations for Figure 22. It shows the correlation between RMSE of CD-SVD, SD-SVD, and CD-CCA. Here, error of CD-SVD and SD-SVD are highly and positively correlated, while CD-CCA has an insignificant correlation with them.

### 5.5 PERFORMANCE ANALYSIS OF ALGORITHMS

In this section, we focus on the performance and running time of the proposed and baseline algorithms. Then, we report the running time of each of the algorithms on the Supermarket dataset to compare them in this aspect.

In CD-CCA algorithm, we compute the canonical correlation between two domains, multiply the projection of source domain (canonical variates) with the diagonal correlations matrix, and project it back to the target space by multiplying the results with the discovered components for the target domain. The complexity of calculating CCA using the approach presented in [44] is $\mathrm{O}(Nk(3n+5m+2mn))$, in which $N$ is the number of iterations for least squares, $k$ is the number of components (equal to or less than the number of items in the source domain), $n$ is the number of datapoints (users), and $m$ is the number of items in the target domain. The complexity for multiplying the $n \times k$ canonical variate matrix of the source domain, to the diagonal $k \times k$ matrix of canonical correlations is $\mathrm{O}(nk)$. Lastly, projecting the target domain canonical variates back to the original target domain space costs O($nkp$), in which $p$ is the number of items in the target domain. Thus, since we have $k < m$ and $k < p$, the complexity of CD-CCA algorithm is O($Nk(3n + 5m + 2mn) + nkp$).

In the large-scale CD-LCCA, the complexity for computing canonical correlations includes iterations of LING least squares algorithm and QR-decomposition of projection of original source and target matrices into their small-scale versions. Ling costs O($np(N_2+k_{pc})$) in each iteration, which $N_2$ is the number of iterations to compute $Y_r$ in large-scale CCA using gradient descent; and $k_{pc}$ is the number of singular values that are used for calculating $U_1U_1^TY$. Each QR-decomposition takes O($nk^2$), in which $k$ is the number of components. Eventually, calculating large-scale CCA will cost O($Nnp(N_2 + k_{pc}) + Nnk^2$). Since we are using sparse matrices in Matlab the multiplications in CD-LCCA depend on the number of nonzero elements in the matrices. In the worst case of multiplying dense matrices, the multiplications will cost O($npk + nk^2$). Thus, as a whole, CD-LCCA will cost O($Nnp(N_2 + k_{pc}) + Nnk^2 + npk$).

Among the baseline algorithms, SVD++ is the fastest. Since it is implemented for sparse matrices, its complexity depends on the number of nonzero elements in the matrix. So, if $|R_u|$ shows the number of ratings by user $u$, the complexity for SVD++ is $O(\Sigma_u|R_u|^2)$.

Figure 24 shows an example of running time of CD-CCA on different domain pairs in the Yelp dataset. The $X$ axis shows the size of domain-pair based on number of items and users. It is in the logarithmic scale and represent the sum of user-item rating matrix sizes in the source and target domains ($\log_{10}(nm + np)$). The $Y$ axis shows the running time of CD-CCA in seconds in logarithmic scale. We can see four examples of domain pairs in the picture. As we can see, as the size of domain-pairs grow, the running time of CD-CCA increases respectively.

To have an analysis of algorithms' performance in practice, we report a sample running time on one of the datasets. We ran all of the algorithms on two similar machines: a MacOS machine with 64GB RAM and two 4-core Intel Xeon, 2.26GHz CPUs and a Linux machine (CentOS) with 64GB RAM and two 4-core Intel Xeon, 2.40GHz CPUs. For CD-CCA, RMGM, and CMF, we use Matlab platform and for CD-SVD and SD-SVD, we use GraphChi software. The average running time of each algorithm on one domain pair of the Supermarket dataset is listed in Table 15. As we can see, CD-CCA has the least running time

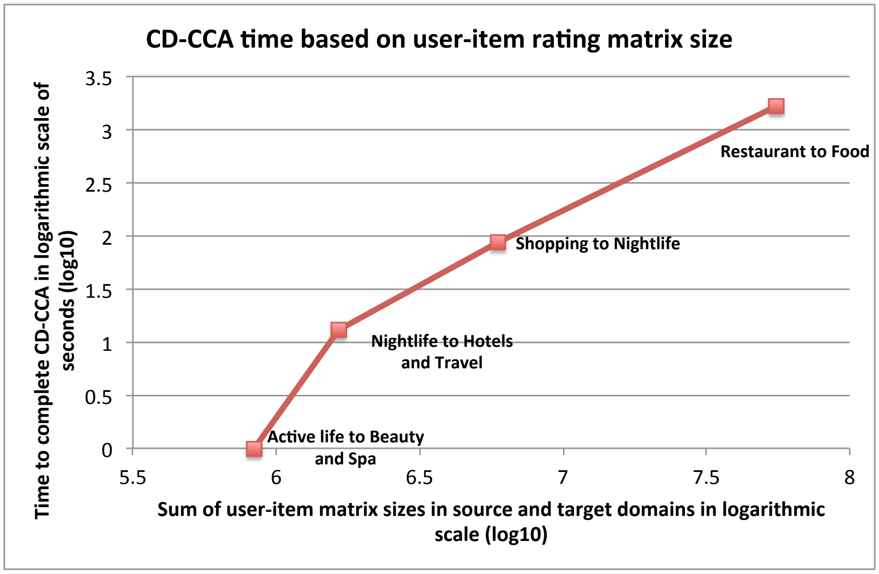


Figure 24: CD-CCA running time in four sample domain-pairs of the Yelp dataset. Numbers are in logarithmic scale.

and RMGM is very slow compared to the other algorithms. One reason for fast running time of CD-CCA is that it can be implemented in full matrices in Matlab and we can avoid loops in its implementation. However, the large-scale implementation of CD-CCA (or CD-LCCA) needs to work with the sparse matrix format in Matlab, and thus, uses less memory and is slow. Running CD-LCCA in Matlab on one domain pair of the Imhonet dataset took 21210 seconds (close to 6 hours) on average. Running CD-SVD with GraphChi on one domain pair of same dataset took almost 4 hours on average.

## 5.6 SUMMARY

In this chapter of the dissertation, we experimented on different, cross-domain and single-domain, algorithms on three datasets with various characteristics. We studied the feasibility and benefits of cross-domain recommender algorithms, including our proposed algorithms, CD-CCA and CD-LCCA.

Table 15: Average running time of each algorithm on one domain pair in the Supermarket dataset

| | CD-CCA | CD-SVD | SD-SVD | RMGM | CMF |
|---|---|---|---|---|---|
| Running time (s) | 36 | 252 | 176.4 | 11224 | 295.38 |

We compared the results of algorithms in each of the datasets and concluded that CD-CCA is the best performing algorithm in the Yelp and Imhonet dataset, and RMGM is the best-performing one in the Supermarket dataset. On the other hand, RMGM is the worst-performing algorithm in the Yelp dataset. One of the reasons that can result in this inconsistency is the characteristics of the datasets. As we have discussed in Section 3.3.2, RMGM algorithm has problems in finding clear clusters of users and items in case there is a high skewness in the ratings of a dataset. If we look at the skewness of ratings in the Yelp dataset, we can see that most of the ratings in the Yelp dataset are on the popular items. The high skewness of the ratings in the Yelp dataset and low skewness of them in the Supermarket dataset can be one of the reasons for this inconsistency. In general, rating-based recommender systems, such as Imhonet and Yelp, are more prone to be naturally skewed; while in the recommender systems based on “implicit feedback” we see more balance in the feedback on items. Also, the nature of Supermarket dataset, in which we have the whole data on the purchased items, is inherently different from the other two datasets. Because, in Yelp and Imhonet datasets, we do not have access to the “consumption” data, e.g. we do not know if a user has gone to a restaurant or not. We only have the rating information of users, if they decide to rate the item that have consumed.

Another reason can be because of the way we processed the Supermarket dataset for RMGM. As mentioned in Section 4.3, we had to convert the frequency of purchases to a categorical rating for RMGM. Although we have lost some of the precision of data because of this pre-processing, the 10-scale categorization in the Supermarket dataset provides more flexibility compared to the 5-Likert scale of the Yelp dataset.

The third likely reason, is the sparsity of the Yelp dataset, compared to the Supermarket dataset. As we have seen in Section 4.4, most of the domains in the Supermarket dataset are denser compared to the Yelp dataset domains. Again, the sparsity problem often happens more in the rating-based datasets compared to the implicit feedback ones. Because, in the explicit rating feedback, the data passes through another cognitive decision of the users, e.g. to decide if they would like to rate the items or not. While, in the implicit feedback datasets, we only see the first cognitive decision of users: to consume (purchase) the item or not. We have mentioned in Section 3.3.2 that RMGM has a poor performance in very sparse datasets.

Also, we have seen that CMF is one of the best-performing algorithms on the Yelp dataset and the worst-performing one in the Supermarket dataset. In both of the datasets, CMF has the most variance of error, and thus widest confidence intervals. We hypothesize that the reason behind CMF's inconsistency of performance is the ratio between number of users and number of (target) items in the two datasets. As we have seen in Section 4.4, in the Yelp dataset most of the domains have a tall user-item matrix. However, the user-item matrices in the Supermarket dataset are usually fat. Since CMF is trying to find a common user factor matrix between the source and target domains, the flexibility of item factor matrices of these domains decreases. Consequently, this leads to better representation of items when there are fewer number of items to fit in the item factor matrix.

Another interesting observation is the correlation among the errors of algorithms. We can see that in all of the datasets, if there is a significant correlation between the error of two cross-domain algorithms, this correlation is positive. However, the correlation between error of SD-SVD and other algorithms varies between the datasets. In the Supermarket dataset SD-SVD's error has a positive correlation with error of CD-SVD; and a negative one with the rest of cross-domain algorithms' errors. In the Imhonet dataset, there is no significant correlation between error of CD-SVD and CD-CCA. In contrast, SD-SVD error's correlation with all of the cross-domain algorithms in the Yelp dataset is positive. This hints us to the effects that the datasets can have on performance of cross-domain algorithms: cross-domain algorithms perform worse where single-domain algorithms perform better in the Supermarket dataset; but, in the Yelp dataset, this relationship is reverse.

We analyzed the time-complexity of the proposed algorithms and compared their running-time with the baseline algorithms. We concluded that CD-CCA is the fastest algorithm on the average-sized data. SD-SVD and CD-SVD are the next fast ones and CMF is slower than these two algorithms. Among all of the algorithms, RMGM is very slow. On the large-scale dataset, CD-SVD and SD-SVD are faster than CD-LCCA. However, the running time of CD-LCCA is reasonable given the size of the data. On the other hand, CMF and RMGM are very slow on the large-scale dataset. Thus, using these two algorithms in large datasets is not practical. Thus, although RMGM performed better than CD-CCA in terms of estimated error in one of the datasets, it may not be practical to use it in large datasets because of its time performance.

In summary, the goal of this chapter was to answer to the first part of our first research question (Q.1.1); to understand if the benefit gained from cross-domain recommenders is because of the extra data, the better algorithm, or both.

We have seen that cross-domain algorithms mostly perform better than, or similar to the single-domain algorithm. In all of the $158 + 50 + 12 = 220$ domain pairs from the three datasets, in only 8 cases SD-SVD performed significantly better than all of the cross-domain algorithms. These 8 domain pairs were all part of the Supermarket purchase dataset. In the rest of the domain pairs, there were at least one cross-domain algorithm that performed significantly better than, or similar to SD-SVD.

Nevertheless, we have seen that cross-domain recommenders do not always increase the quality of recommendation results. In some cases, the cross-domain recommender algorithms did not improve the results, compared to the single-domain algorithm; they just did not have a significantly worse results compared to SD-SVD.

Eventually, we conclude that cross-domain recommender systems are feasible and can be beneficial in some of the domain pairs and datasets.

Also, we have seen that the benefit of these recommender systems, compared to the single-domain recommender, comes from both the additional data available to them and the approach they use to utilize this additional information. CD-SVD algorithm, which uses the cross-domain setup and the single-domain approach, has performed significantly better than SD-SVD in some of the domains of all of the datasets. We attribute this behavior to the extra information that CD-SVD had compared to SD-SVD. However, we have seen that in many cases that SD-SVD performed significantly better than CD-SVD, the other cross-domain algorithms outperformed SD-SVD. In these cases, the additional information alone is not enough to produce better recommendations. But, having better approaches that efficiently use this extra information, results in less error and better recommendations.

In later chapters, we explore the conditions, which lead to better performance of cross-domain recommender systems, compared to the single-domain ones.

# 6.0 COLD-START EXPERIMENTS

One of the major problems in the recommender systems literature is the cold-start problem [56]. For example, in collaborative filtering, the cold-start problem happens when a new user joins the system. Since there are no ratings available for this user, there is no way to compare this user to other users and find out their similar tastes. Thus, the recommender system cannot recommend any items to this new user. One of the goals of cross-domain recommendation is to alleviate the cold-start problem by transferring user information from the source domain to the target domain. In this case, if user is new to the target domain, but has an established profile in the source domain, cross-domain recommender can adjust the target user's source profile for using in the target domain. In CD-CCA, we transfer user profiles from the source domain to the target domain using the canonical variates and canonical correlation that are estimated by CCA.

Since tackling the cold-start problem is one of the main goals of cross-domain recommenders, we compare CD-CCA and the baseline algorithms in the cold-start setting for each of the datasets. Another aspect that can affect the performance of algorithms is users' source domain profile size. It is important to understand how much data is needed to be transferred from the source domain to the target domain to have a reasonable increase in recommender system's performance.

This chapter aims to experiment on the cold-start setting to answer the research question Q1.2. The results of our experiments on each of the datasets in the cold-start setting are presented in the following sections.

## 6.1 EXPERIMENT SETUP

Since we have multiple domain-pairs in each of the datasets, we run the cold-start experiments in two settings: once for each of the domain-pairs separately, and once averaging the errors over all of the domain-pairs.

To understand how each of these algorithms perform in the cold-start setting, we group the test users of each dataset based on the size of their target domain profile. Then, we calculate the error for each group of these users. In the case of analysis of cold-start results for each of the domain pairs separately, we calculate the average user-based error for all of the users with the same target profile size and report that average. For analyzing all of the domain pairs together, we average over user-based error for all of the test users that have the same target profile size, independent of the domain-pair they are coming from.

To study the effect of source profile size on the results, we partition test users based on the size of their source domain profiles and calculate the error for each group separately; once for each of the domain pairs, and once for all of the domain pairs at the same time.

To calculate the user-based error, we calculate the RMSE and MAE of algorithms for each row of the estimated user-rating matrix. So, instead of averaging the error over all of the test ratings, we calculate and average error for each user, based on that specific user's test ratings.

## 6.2 COLD-START ANALYSIS FOR THE SUPERMARKET PURCHASE DATASET

First, we look at the different profile sizes in the Supermarket dataset. To have a dataset-wide view (instead of a domain-pair specific one), we put all of the test users of all domain-pairs together and plot the size of their target profile sizes. Looking at the scatter plot of number of users versus target profile size in Figure 25, we can see that most of the test users have a small target profile size. The maximum number of items in target profiles of test users is 45.

To understand how each of the algorithms perform in the cold-start setting in all of the

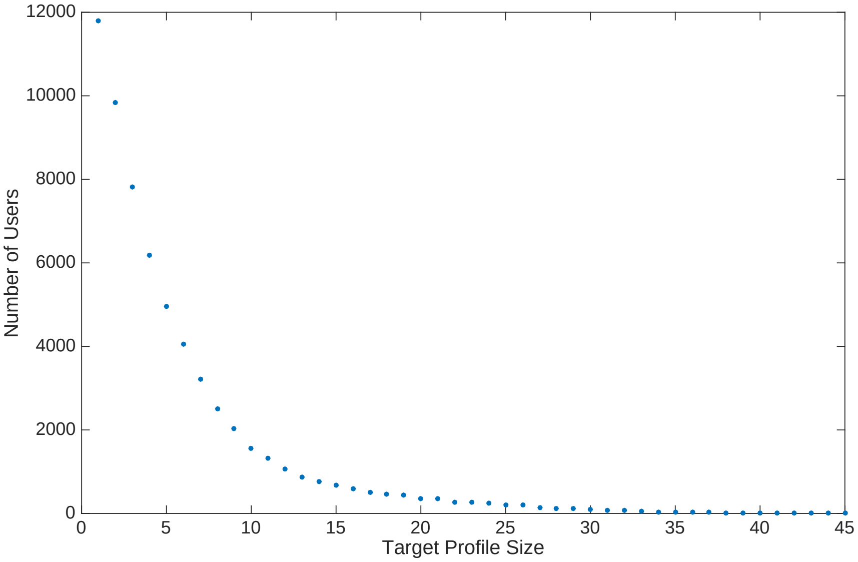


Figure 25: Target profile sizes of users in the Supermarket dataset

domain-pairs, we look at their RMSE and MAE based on users' target profile size. To do so, we calculate the error of each algorithm for each of the test users in each of the domain pairs. Then, we group the test users of all domain pairs based on their target domain profile sizes. For each group of users, we average the error of that group and calculate the 95% confidence interval for that average.

Figures 26 and 27 show the RMSE and MAE of each of the algorithms for test users in all of the domain pairs based on their target domain profile size. As we can see in this figure, the confidence interval for the errors increase as the size of target domain profile grows. This is because there are less number of users with higher target profile sizes in the dataset. Also, we can see that all of the algorithms follow a similar trend of error as users' target domain profile size grows.

Except for RMGM that has a high error when target profile size equals to one, starting from target profile size 1 to around 7, we see an increase in the error rate. Although, the error difference of consequent profile sizes is very small, this difference is significant in many cases, especially for the SD-SVD algorithm. It appears that having more items in the target domain's profile results in more error in all of the algorithms. Since this trend is happening for SD-SVD more significantly than the cross-domain algorithms, we cannot attribute it to the characteristics of cross-domain recommenders.

After the seventh item in target domain's user profile, we can see a decrease in error of cross-domain algorithms. For SD-SVD the error almost stays the same and then increases after the 25 target profile size. CD-CCA, CMF, and RMGM show a more steady error reduction compared to the single-domain algorithm. This error reduction is most visible in the RMGM algorithm's results. While RMGM works worse than all other algorithms for users with two to seven items in their target profiles, it improves very fast for user profile sizes of around 20. More specifically, for these target domain profile sizes, it has a significantly lower RMSE compared to SD-SVD and CD-SVD and significantly lower MAE compared to CD-SVD, SD-SVD, and CD-CCA. After that, RMGM's error increases again. For CD-CCA and CMF, there is a slight, but significant and steady error reduction by increasing users' target domain profile sizes until the profile size of 35. After that, we can see a small increase in the error rates of all algorithms.

Comparing the confidence intervals of errors in these algorithms, we can see that CD-CCA has smaller confidence intervals and shows a more steady behavior. Except for the error of profile size of one, RMGM also has a relatively steady behavior. CMF and CD-CCA have wider confidence intervals and thus are less reliable, especially with large target profile sizes.

The aforementioned results are for average of test users in all of the domain-pairs. We look at each of the domain-pairs separately to get a closer look at the cold-start setting and how each algorithm handles it. Figure 52 in Appendix A shows the MAE and RMSE of each of the algorithms in each of the 50 domain-pairs for different target domain user profile sizes. As we can see in these pictures, the results of many domain-pairs are similar to the average results over all of the domain-pairs. However, in some of the domain-pairs, we can see different trends. For example, for "home cleaning → fruit & vegetables", we can see that CMF has a significantly better RMSE compared to all other algorithms, for users with target profile size of one; and in "home outdoor → fruit & vegetables" domain in the same figure, CD-CCA has a similar RMSE compared to SD-SVD and CD-SVD; and SD-SVD performs

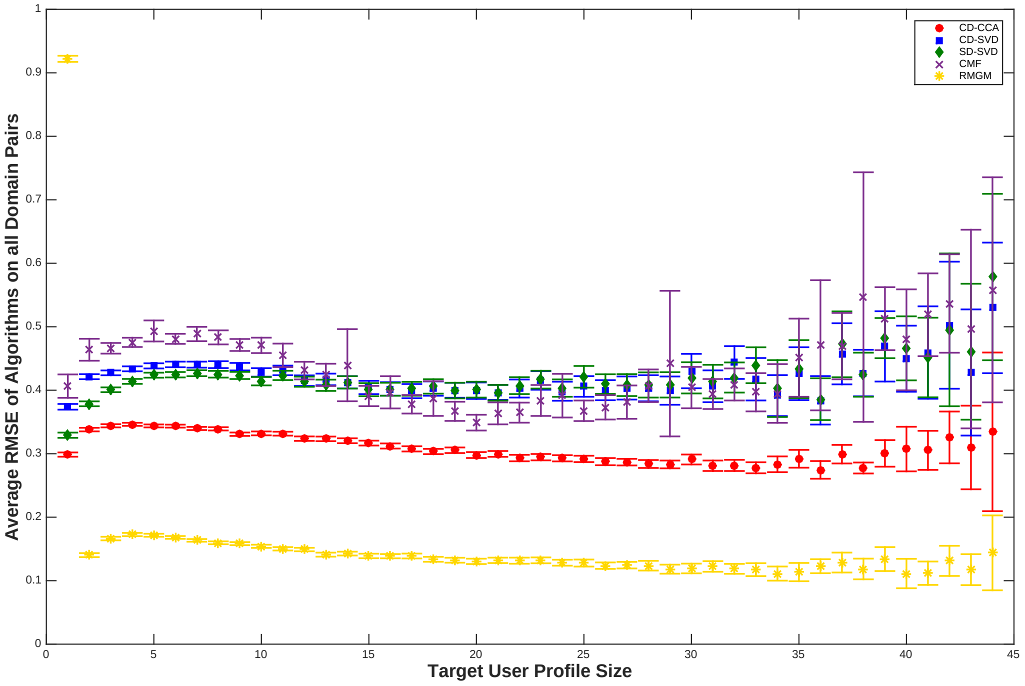


Figure 26: User-based RMSE of algorithms in the Supermarket dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size

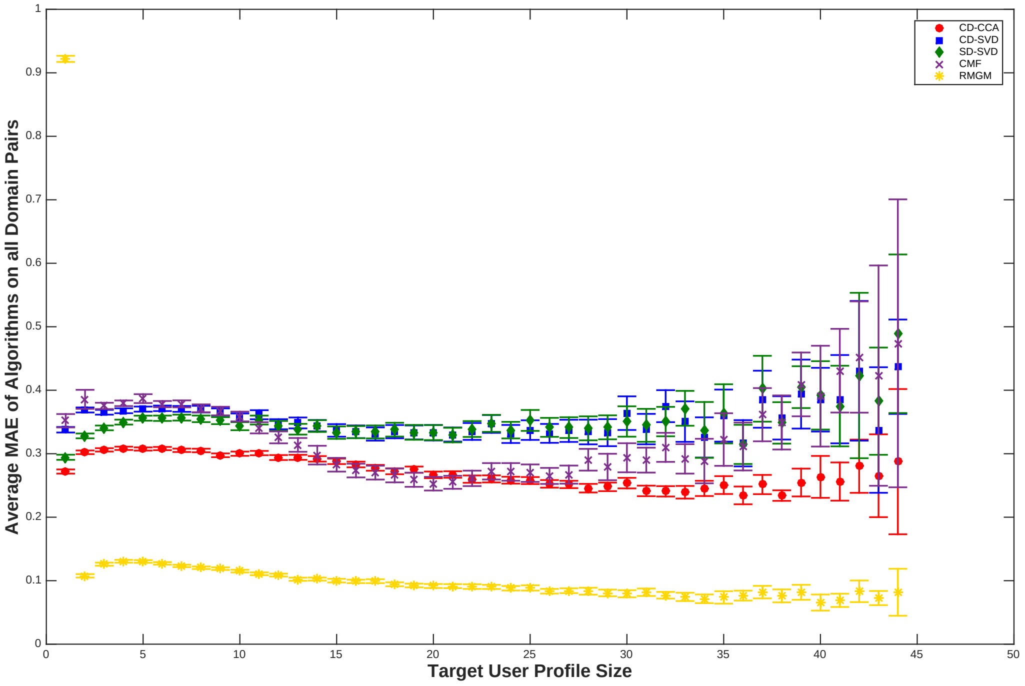


Figure 27: User-based MAE of algorithms in the Supermarket dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size

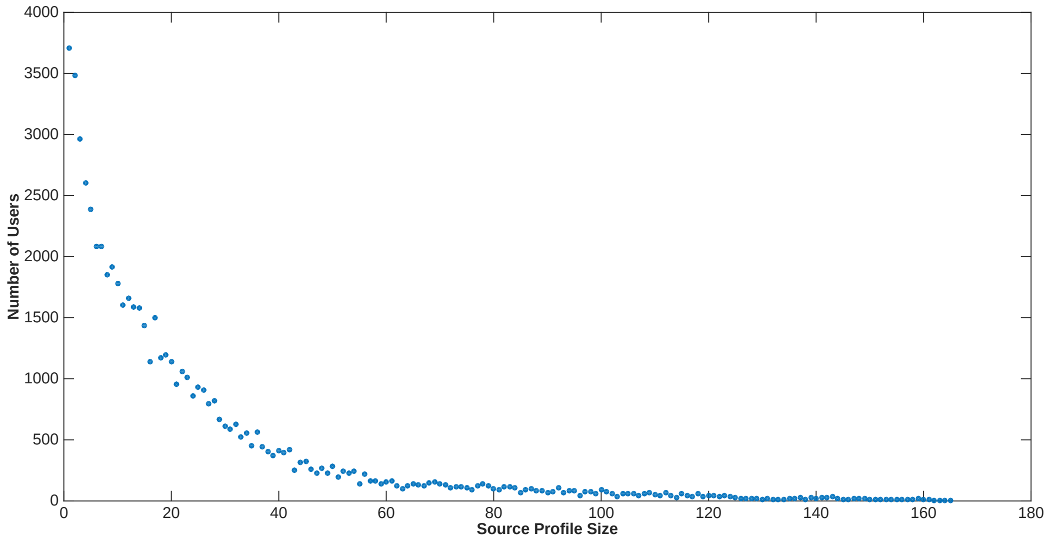


Figure 28: Source profile sizes of users in the Supermarket dataset

mostly better than CD-SVD in "bread → home cleaning". However, although these errors appear to be in a different order than the general results, they do not contradict them. This happens because of different distribution of target domain profile sizes in each of the domain pairs.

Another factor that can impact the results is the size of user profile in the source domain. Figure 28 shows the number of users in all domain-pairs with various source domain profile sizes. The number of items bought by users in the source domains ranges between one and 165. It is important to know if transferring just a few items from an auxiliary domain can help or not. Also, we would like to know if transferring more information from the source domain could harm the recommendation results or not. To understand this, we run the same cold-start experiments on users' source domain profile size.

Figures 29 and 30 show the RMSE and MAE of algorithms, grouped by source domain profile size of users, averaged over all domain pairs. Although SD-SVD does not use the information from any source domains, and thus it should not show a change of error based on source user profile size, we still plot the RMSE of this algorithm to show the changes that

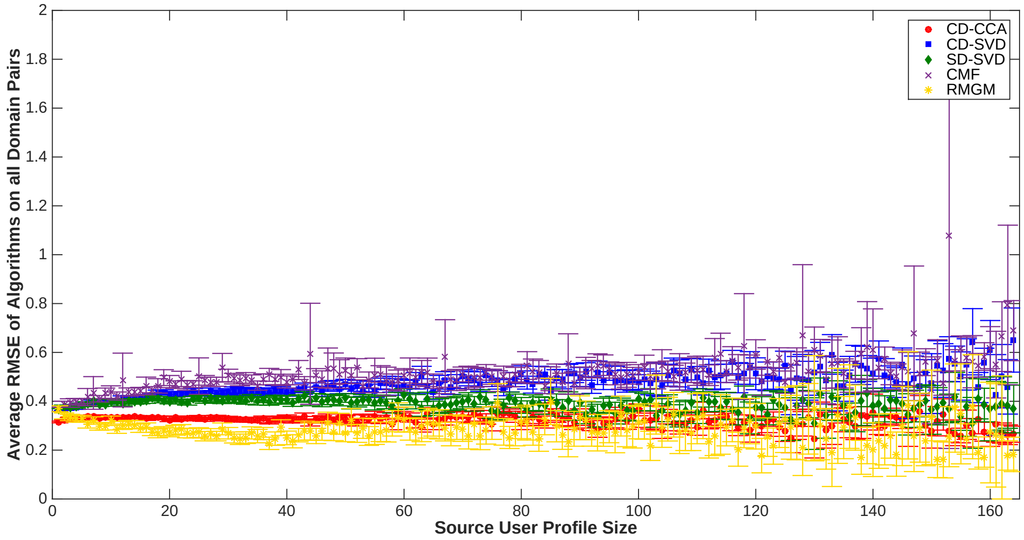


Figure 29: User-based RMSE of algorithms in the Supermarket dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size

happens in other algorithms' errors in correlation with SD-SVD's errors. We can see that the error of SD-SVD, and thus CMF, CD-SVD, and CD-CCA, has a slight increase as the source user profile size grows. This increase is more visible in CMF and CD-SVD compared to CD-CCA and SD-SVD. After having around five items in source profile, CD-CCA's errors start to decrease. For CD-SVD and CMF, the increase will continue by users' source domain profile size. This trend is especially visible in the MAE of algorithms. For SD-SVD, the errors stay more or less the same. For RMGM, we see a steady decrease of error from the beginning until 40 items in users' source domain profile. After having around 40 items in users' source profile, the variance of errors increases and we cannot rely on any increasing or decreasing error trends.

Another interesting observation is that CD-CCA performs better than all of the other algorithms when having a very small source profile size. After having about 5 items in the source domain profile, RMGM has the best error among these algorithms.

To understand the effect of source domain profile size on the errors we can look at Figure

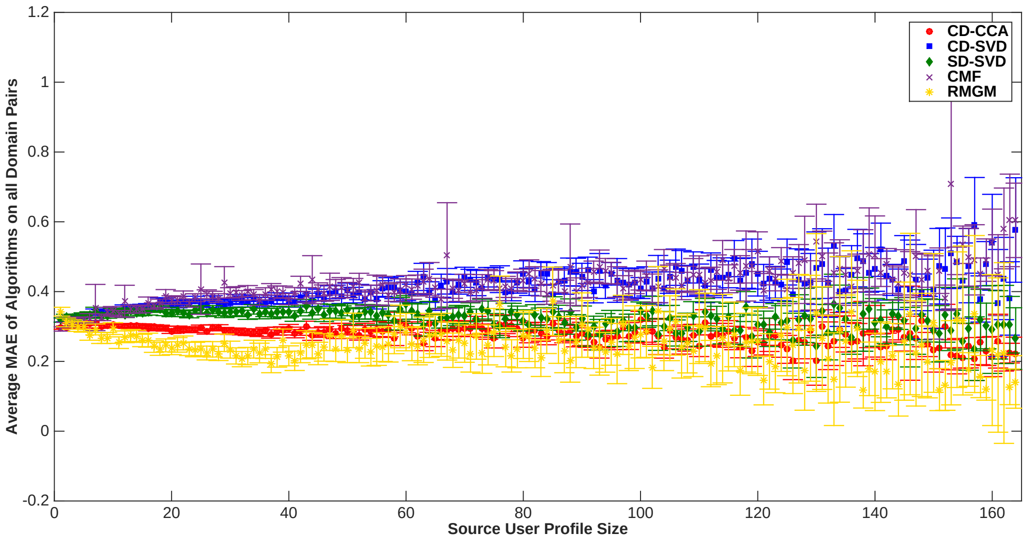


Figure 30: User-based MAE of algorithms in the Supermarket dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size

54 in Appendix A. Looking at these figures, we can see some different error patterns in each of the domain-pairs compared to the average error patterns for all domain pairs. For example, CMF is one of the best-performing algorithms in the "canned & pickled → fruit & vegetables" domain pair; RMGM has the most error for the "bread → events" domain pair; and MAE of CD-CCA in "home cleaning → fruit & vegetables" decreases with a large steep until the biggest source domain profile size.

## 6.3 COLD-START ANALYSIS FOR THE YELP DATASET

Looking at the target domain profile sizes of users in Figure 31, we can see the rapid drop in user profile sizes. Most of users (92% of them) have only one to three ratings in a domain, and very few of them ($8.9219e-07\%$) have more than 45 item ratings in their target profile. This rapid drop in user profile sizes results in a more severe cold-start problem in the Yelp

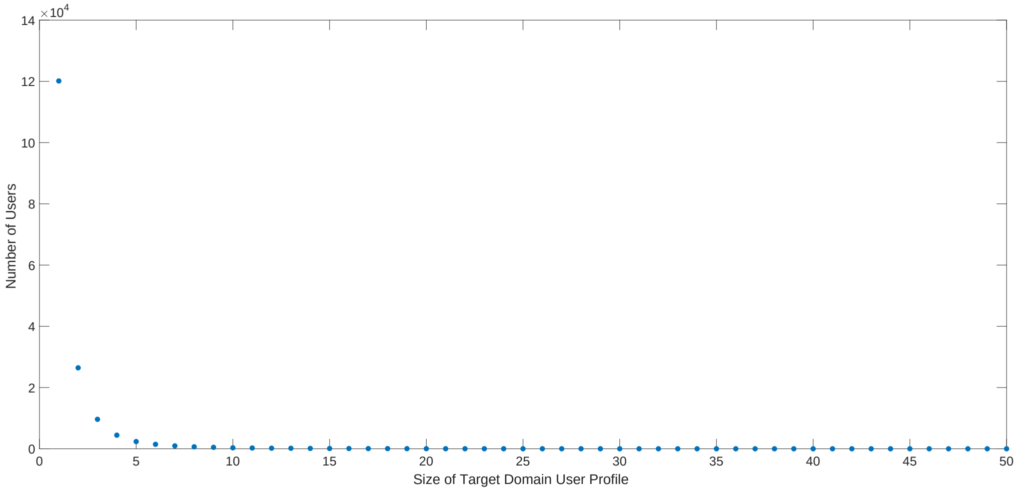


Figure 31: Target profile sizes of users in Yelp dataset

dataset compared to the Supermarket dataset.

We run the cold-start experiments on the Yelp dataset in a similar setup to the Supermarket dataset, as explained in Section 6.2. First, we look at the average performance of algorithms on all 158 domain pairs, based on test user profile sizes in target domains. Figure 32 shows the RMSE of algorithms in the cold-start setting and Figure 33 shows their performance based on MAE.

Based on these figures, averaging over all of the domain pairs, CD-CCA is performing the best in the cold-start setting; and CMF is the next best algorithm. CD-SVD and SD-SVD were unable to return recommendations in the extreme cold-start situation, where we have up to four items in the target user profile. RMGM has a large error when the test users have only one item in their profile. However, after that, its error drops dramatically and continues to decrease as the target profile of user grows in size. For CD-SVD and SD-SVD, there is a slight decrease in the error as the user profile size increases. However, both CMF and CD-CCA experience a small increase in the error until target profile size of three.

As we can see in the pictures, as the user profile size grows, so does the confidence interval of the error. This is because of the small number of users with a larger profile size. Thus,

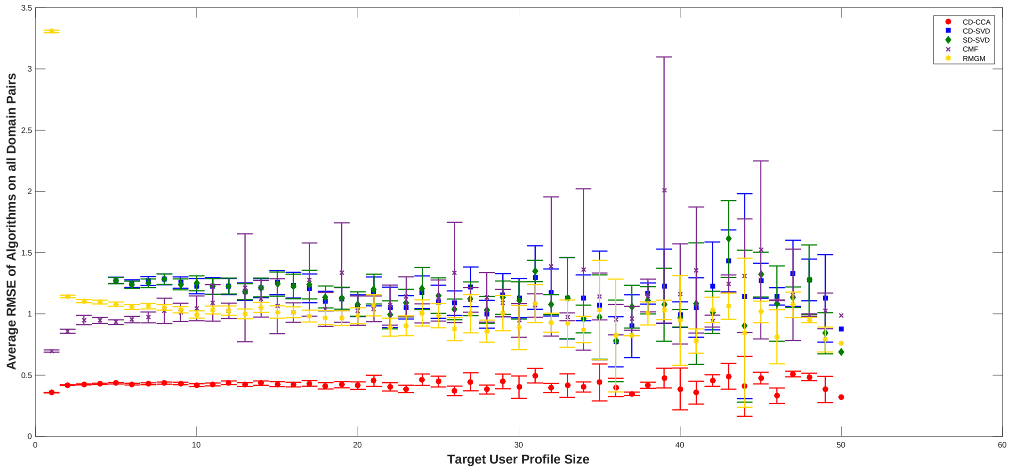


Figure 32: User-based RMSE of algorithms in the Yelp dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size

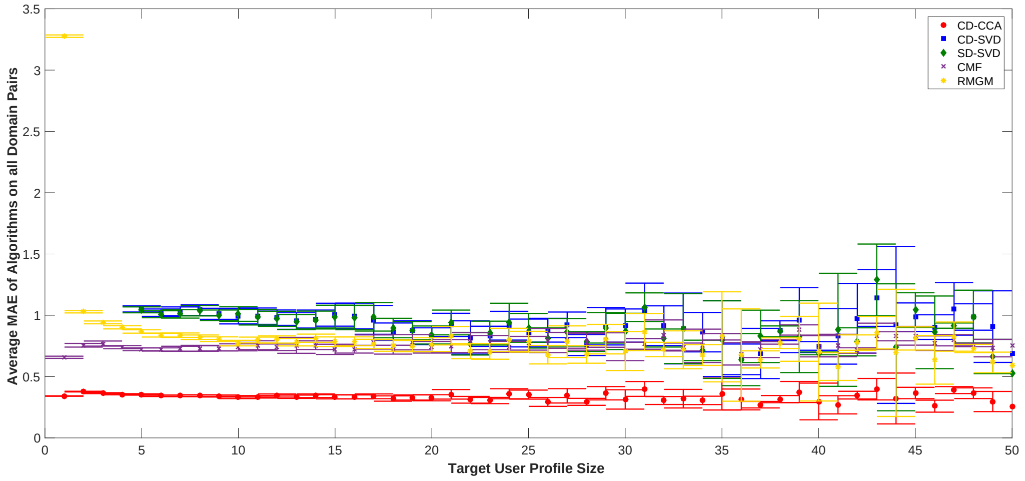


Figure 33: User-based MAE of algorithms in the Yelp dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size

the reliability of results decreases for the errors reported at larger target profile sizes.

We look at the errors reported by each of the algorithms in each of the domain-pairs separately. Figure 74 in Appendix B.3 shows the MAE of algorithms in each of the domain pairs[1].

The first point that is noticed by looking at these pictures is the size of user profiles in various domain pairs. As we can see, in only 26 of the domain pairs the maximum size of user profiles exceeds 10 items and in only 7 of them this size is bigger than 20 items. The domain pairs with largest test user profile sizes are "food → restaurants" and "night life → restaurants" with largest target profile size of 50. As a result, the results that we see in the tail of plots in Figures 33 and 32 are generated from these domain pairs. In some domain pairs, such as "public services → financial services" and "religious organizations → education", the target domain user profiles have less than three items. Consequently, since SD-SVD and CD-SVD were not able to generate recommendations in many extreme cold-start situations, we can mostly see the results of CD-CCA. CMF, and RMGM in these domain pairs.

Although there are profile size differences in the domain pairs, most of the algorithms show a similar performance to the one calculated as average of all domain-pair user profile sizes. There are a few cases in which the algorithms show a different behavior compared to the average case. For example, in "professional services → financial services", there is no significant differences between CD-CCA, CMF, CD-SVD, and SD-SVD errors; in "night life → food" and "arts & entertainment → night life", the error rates of SD-SVD and CD-SVD are much higher than CD-CCA, CMF, and RMGM; and in "active life → beauty & spas", CD-SVD's MAE is slightly, but constantly, higher than SD-SVD.

In the next step, we perform an analysis on users' source domain profile size. Figure 34 shows the number of test users with various profile sizes in the source domain. The maximum ratings a test user has in the source domain is 145 items. However, more than 92% of users have a source profile size of 10 items or less. To see the effect of source domain profile size on the recommendation results, we look at the RMSE and MAE of all algorithms in Figures 35 and 36. These figures show the error of algorithms, for users of various source domain

[1] We have omitted the figures for RMSE of the algorithms because of the large number of domain pairs.

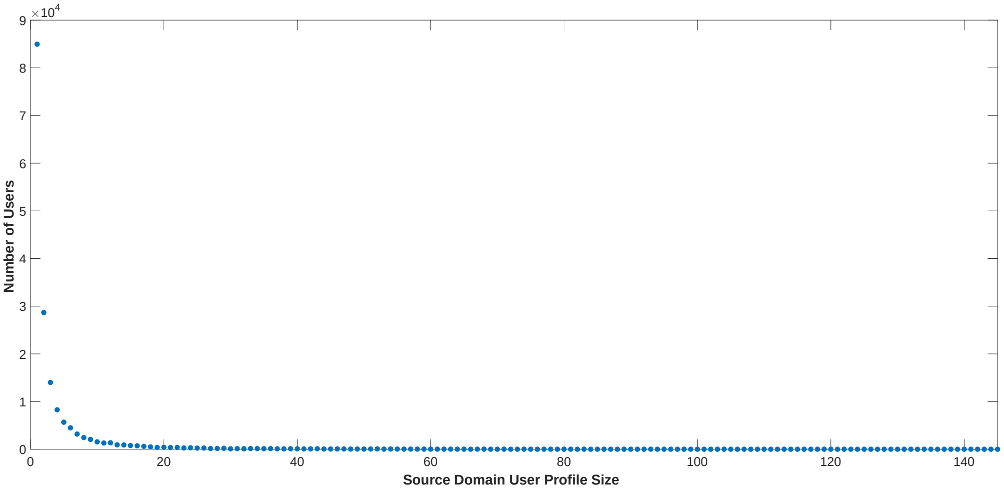


Figure 34: Source profile sizes of users in Yelp dataset

profile sizes, averaged on all domain pairs.

In these figures, we can see that the error of RMGM algorithm decreases as users' source profile size increases. For CD-CCA, SD-SVD, and CD-SVD, the error change is insignificant. However, for CMF, there is a slight increase in the errors as users have more items rated in their source domain profile.

The confidence interval of all errors increases with the increase of user profile size. This increase may be because of the less number of users that we have with larger profile sizes. This increase in confidence interval is more obvious for RMGM and CMF. These two algorithms produce less stable errors compared to CD-CCA, CD-SVD, and SD-SVD.

## 6.4 COLD-START ANALYSIS FOR THE IMHONET DATASET

To understand how CD-SVD, SD-SVD, and CD-CCA perform in the cold-start setting in the Imhonet dataset, we look at the target domain profile sizes of users. Figure 37 shows the number of test users with each of the target domain profile sizes in all of the domain

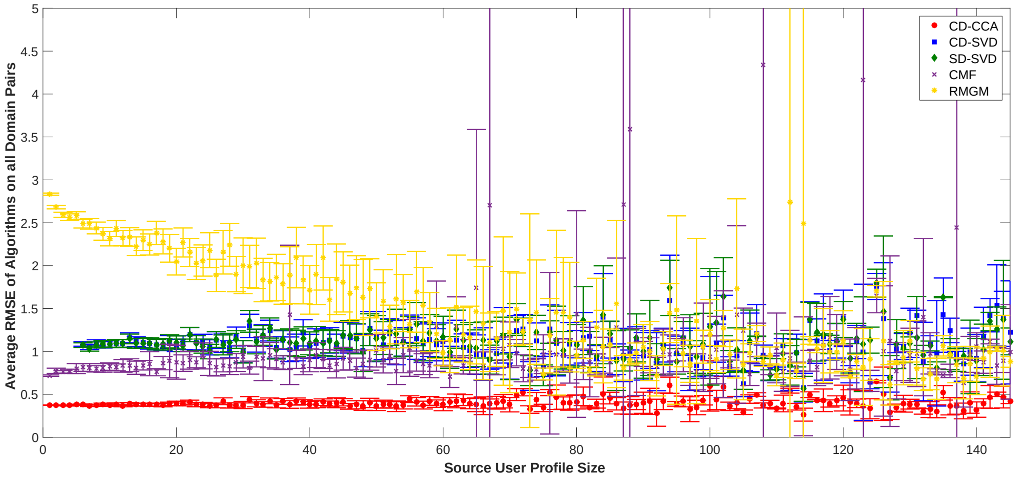


Figure 35: User-based RMSE of algorithms in the Yelp dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size

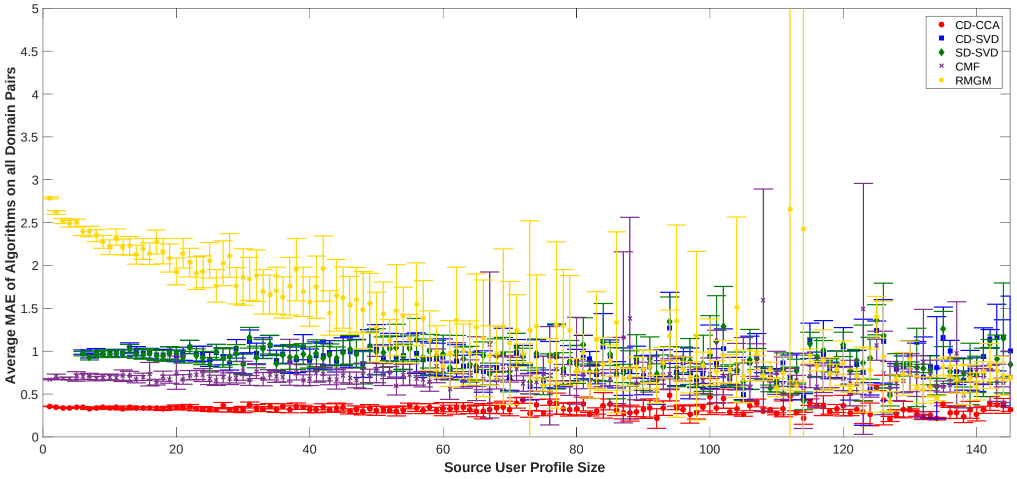


Figure 36: User-based MAE of algorithms in the Yelp dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size

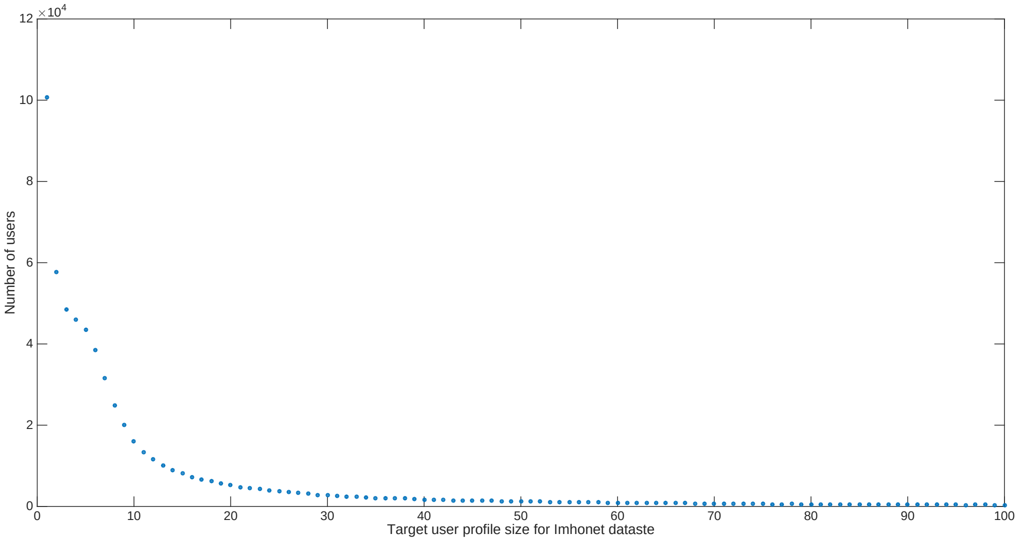


Figure 37: Target profile sizes of users in Imhonet dataset

pairs. We can see that most of the test users have a small profile size (less than 10 items) in the target domain. There are a few users with 100 and more items in their target profile. However, to have a better plot, we skipped showing these users. Also, we can see a concave shape at the small (less than 10) target domain profile sizes. This happens due to the data collection procedure in Imhonet. To collect more data from users, Imhonet has asked some of the users to rate at least 20 items, so that Imhonet can provide recommendations to them. Since we only use 20% of test user ratings in their target profiles, this increase in the profile size happens for the profiles that have less than 10 items.

For the cold-start experiments in the Imhonet dataset, we follow the same instructions as for the other two datasets. We calculate user-based errors in each of the domain pairs. Then, we average over the error of users with the same profile size in all of the domain pairs. Figures 38 and 39 show the RMSE and MAE of algorithms in the cold-start setting based on target user profile size.

As we can see in the pictures, for all of the algorithms, we see an increase of error as the target profile size grows, until a maximum point of error happens. After that, we can

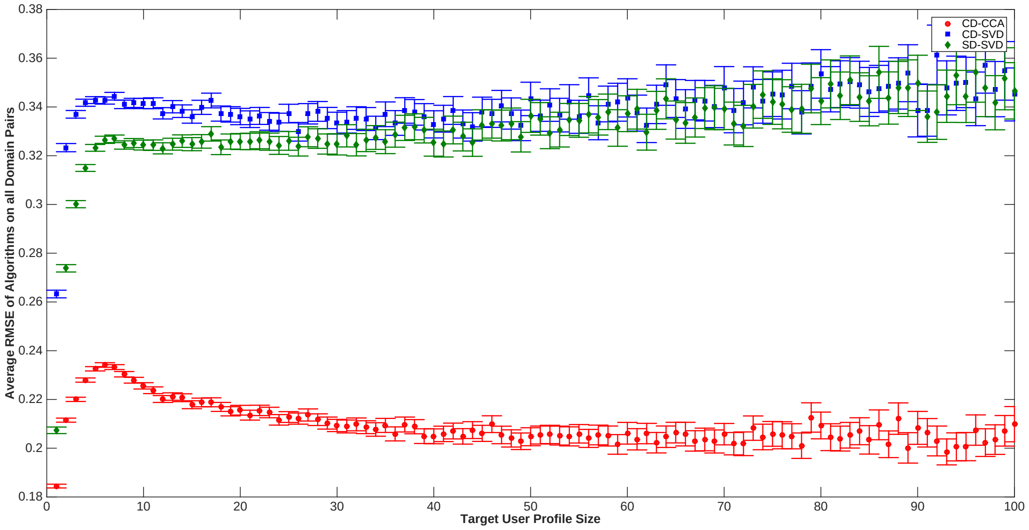


Figure 38: User-based RMSE of algorithms in the Imhonet dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size

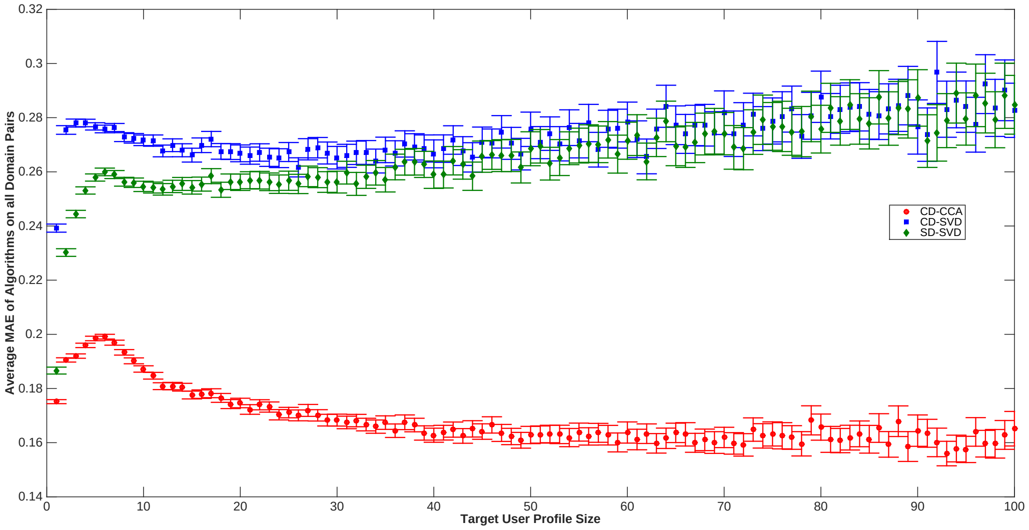


Figure 39: User-based MAE of algorithms in the Imhonet dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size

see a drop in the error rates by increasing target profile sizes. For SD-SVD and CD-SVD, we can see an insignificant increase (or steadiness) of error after a while. However, the error of CD-CCA continues to decrease by increasing size of target domain user profiles. Also, as the target user profile sizes increase, and we have less number of users with larger profile sizes, the confidence interval of error gets wider. CD-CCA has a significantly lower error compared to SD-SVD and CD-SVD for all user profile sizes. SD-SVD has a significantly better performance compared to CD-SVD, up to target profile size of 40. After 40 items, SD-SVD and CD-SVD become comparable in error.

Figures 97 and 98 in Appendix C.3 show the cold-start results of each of the algorithms in each of the domain pairs. As we can see in these figures, CD-CCA is generally performing better than the other two algorithms. But, we have different results, especially for CD-SVD and SD-SVD, in some of the domain pairs. For example, in “book → game”, CD-CCA and the other two algorithms get to perform similar to each other after users have enough items in their target profile (around 45 items); in “book → perfume”, the error is mostly increasing as the target profile sizes grow; in “movie → game”, SD-SVD performs much better than CD-SVD from the beginning, but in “game → movie”, CD-SVD is sometimes significantly better than SD-SVD; and in “movie → book” the error of all three algorithms continue to decrease after a certain point in user profile size.

Also, we can see the difference in confidence of algorithms in different domain pairs. For example, in “book → movie” the errorbars are much tighter than in “game → perfume”.

Studying the same setup for the source domain profile sizes, we look at source domain profile sizes of users in Figure 40. We can see that there is a break in the picture for source domain profile size of 20; the number of users with 20 items in their profile is suddenly higher than the neighboring profile sizes. The reason is the same as for the concave shape of target profile sizes: the data collection procedure in Imhonet. Other than this exception, we can see that the graph has a familiar trend: more users with a few ratings in their profiles and less users with more ratings. There are a few users that have more than 500 item ratings in their profile that we are not showing in this picture.

Experimenting on the source domain profile sizes, results in Figures 41 and 42 for RMSE and MAE of algorithms. We can see that there is a sharp increase of error for all of the

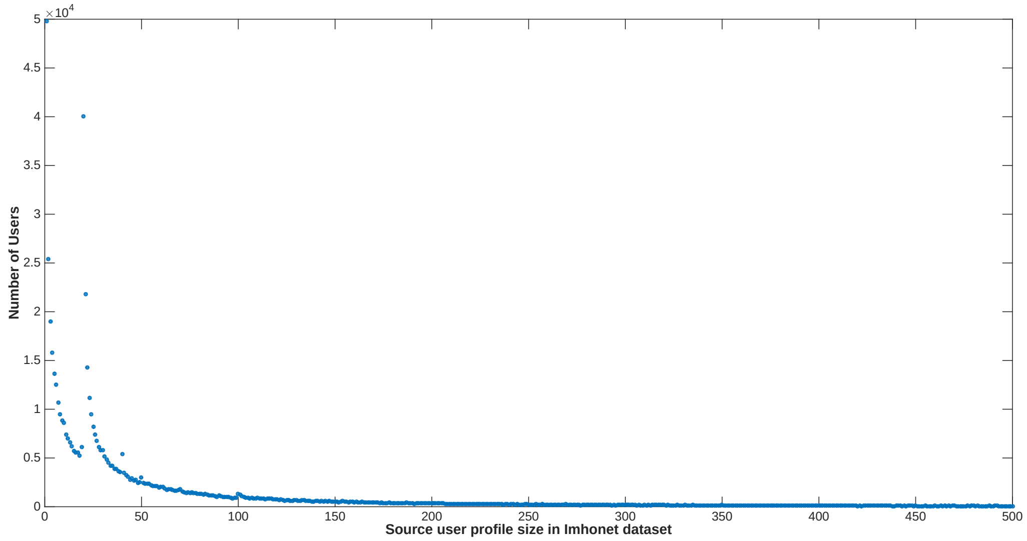


Figure 40: Source profile sizes of users in Imhonet dataset

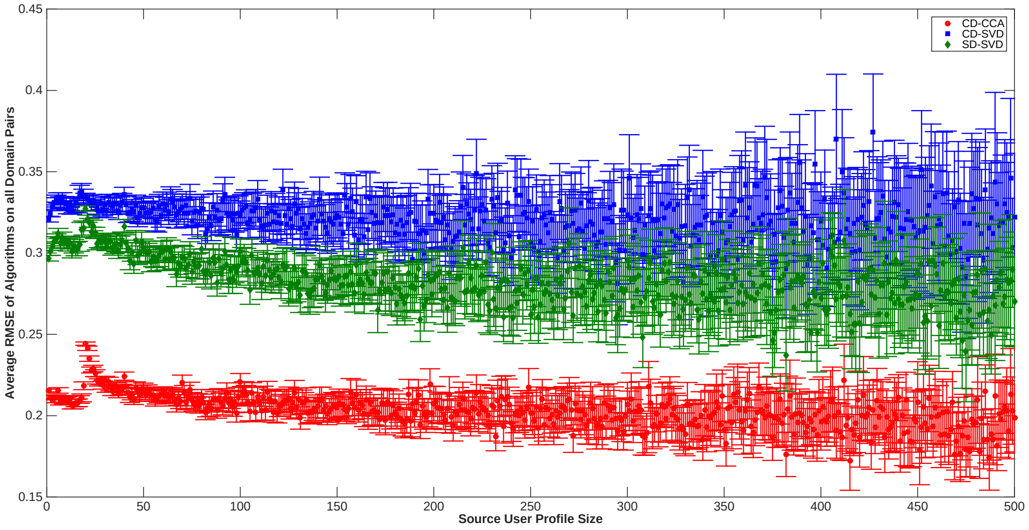


Figure 41: User-based RMSE of algorithms in the Imhonet dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size

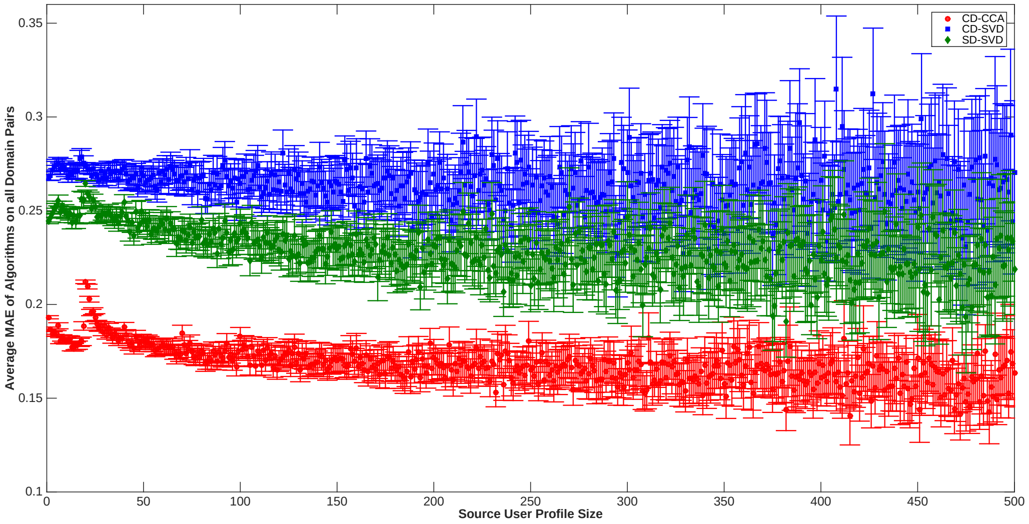


Figure 42: User-based MAE of algorithms in the Imhonet dataset, averaged on all domain-pairs and sorted based on the users' source domain profile size

algorithms at the 20 profile size. This can be because of the number of source profile sizes with 20 items in them. Also, we can see that the error of all algorithms decreases after this point by increase in the source profile size of users. For SD-SVD, this decrease is unexpected, because this algorithm is not using source domain information. However, we speculate that the decrease may be because of correlation of source domain profile sizes with another factor that results in less error for SD-SVD.

Looking at the effect of source profile size on the error of algorithms in Figures 99 and 100 in Appendix C.3, we can see some differences in the cold-start results of different domain pairs. For example, we can see that the algorithms perform more similar to each other in "perfume → game" versus "perfume → book"; CD-SVD is the worst algorithm in "book → game", but better than SD-SVD for smaller profile sizes in "book → movie"; and the confidence of error in "book → perfume" is much more than "game → perfume".

## 6.5 SUMMARY

In this chapter, we experimented on the cold-start setting for different, cross-domain and single-domain, algorithms on three datasets. We looked at the cold-start setting in the target domain profile of users to research on the possible answers for research question Q1.2. We also looked at the source domain profile sizes to see the effect of amount of data that is transferred from the source domain on the recommendation results.

An interesting observation in the cold-start profile is that the average domain errors of almost all of the algorithms in all of the datasets increases at the beginning by the increase of target profile size (despite the fact that we expect a decrease in error because of the increase of information for users). This increase is more visible in the Supermarket and Imhonet datasets and SD-SVD, CMF, and CD-SVD algorithms. For CD-CCA, it only happens briefly, and then the error decreases. We hypothesize that this increase is because there are many domain pairs with a few number of users and very small profile sizes in the datasets. In other words, the maximum profile size of users in these domains are very low. In general, the errors for recommendations in these domain pairs can be high because of the lack of enough overall information, e.g. due to sparsity, to provide good recommendation. This phenomenon results in increasing the average error in the beginning for the target profile sizes. As we have seen in the figures related to each of the domain pairs, this increase does not happen in the domain pairs with larger target profile sizes and more data.

An exception to this reasoning is the CMF algorithm. As the target profile grows, the error of this algorithm grows even in the domain pairs with larger target profile sizes. This is more visible in the Supermarket dataset with denser domains. This can be because of the design of this algorithm: it tries to find a common set of factors between the users in the source and target domains; while having a small item factor matrix for each of the two domains. As the number of items grow, it will be more difficult for this algorithm to fit all of the item features in a small item factor matrix, especially in dense domains with more information about items.

The exception to the initial growth of average error on all domains is the RMGM algorithm on the Yelp dataset. It has a monotonic decreasing trend as the target profile of users grow. As we have mentioned in Section 3.3.2, RMGM has problems to find the shared user-item clusters in skewed and sparse datasets. As the profile size of users grow, the sparsity of the dataset and its skewness decrease and thus results in better recommendations for RMGM. Since RMGM is performing very poorly for users with a very small profile size, no matter how big is the largest profile size of the domain pair, the reasoning that applies to the initial increase of average error in the other algorithms does not apply to it.

After the initial increase of error, we see different trends of errors based on user target profile sizes in different datasets. For the Yelp and Supermarket datasets, we mostly see a decrease of average error on all domain pairs when the target profile sizes exceed a specific size.But, for the Imhonet dataset, the error of SD-SVD and CD-SVD algorithms has an insignificant increase. We hypothesize that this happens because of the extreme sparsity of Imhonet as we have seen in Section 4.4.

As a summary of error changes over the source profile size, we can see that, while there should not be any changes in the error of SD-SVD based on the source profile size, its average error has a small increase or decrease as the source profile size increases in all of the datasets. This relationship can be because of some other variables that change with the source profile size. For example, if users with larger source profile size also have a larger target profile size, the error of SD-SVD will be correlated with the source profile size through its correlation with the target profile size. The average error of CMF and CD-SVD increases as the source profile size grows in the Yelp and Supermarket datasets. This increase means that these two algorithms cannot handle extra (unrelated) information about the users and more source domain information will add more noise and thus harm their performances in average. CD-CCA has a relatively small decrease of error at the beginning and then has a steady or decreasing error as the source domain profile size of users grow. This hints that CD-CCA can use the extra source domain information at the beginning and then, handles the noise that comes with adding too much source domain information. However, RMGM is the best algorithm in handling extra source domain information. Starting with a very high error at the beginning, it seems that RMGM cannot use a small amount of information from the source domain efficiently (especially in the Supermarket dataset). However, as the source domain profile sizes grow, the error of RMGM decreases constantly and with a fast pace.

In summary, we conclude that CD-CCA is the best algorithm in handling the cold-start situation in general. It has a low error in the extreme cold-start setting for target domain; it can use the moderate amount of source domain data to reduce the error; and it can handle the extra source domain information without harming the recommendations.

RMGM is the second best algorithm for the cold-start situation because even though it performs better than other algorithms in some of the datasets and even though it has a decreasing trend of error on having more and more information, it has a very bad performance in the extreme cold-start case of having one or very few ratings in user profiles.

Finally, to answer to Q1.2, we can conclude that cross-domain recommender systems, especially CD-CCA, can be beneficial in the cold-start setting.

# 7.0 FINDING THE APPROPRIATE AUXILIARY DOMAIN

In this chapter of the dissertation, the goal is to find the data characteristics that lead us to a better cross-domain recommendation and a higher improvement in cross-domain recommendations versus single-domain recommendations. Discovering these characteristics can lead us to select the best source domain for a specific target domain before performing the cross-domain recommendation task.

Here, we use CCA, in addition to other data characteristics, as key factors to find the best auxiliary domain for a specific target domain. We use this tool to answer the research questions Q2.1, Q2.2, and Q2.3. We hypothesize that the more canonical correlation the two domains have, the better the performance of cross-domain recommender system will be. To test this hypothesis, we analyze the correlation of the error of cross-domain recommendations with the CCA results, each domain data characteristics, and domain-pair data characteristics.

The second hypothesis in this chapter is on the improvement of cross-domain recommendations, compared to the single-domain recommendations. Since, based on our results in Chapter 5, the cross-domain algorithms' error is correlated with the single-domain algorithm's error, we would like to study if CCA can be a major factor in defining the amount of improvement that can be achieved by cross-domain algorithms over single-domain algorithms.

Eventually, to have a global view of effect of all of the data characteristics, at the same time, on cross-domain recommender results, we perform a regression analysis in the next section.

In the following sections, we first introduce the data characteristics that we use from the datasets. Then, we perform a correlation analysis between these data characteristics and the error of each of the algorithms. After that, we run a regression analysis with these data characteristics as dependent variables and the error of each algorithm as the dependent variable. Finally, we look at the domain pairs to find out if the good domain pair can make sense intuitively.

## 7.1 DATA CHARACTERISTICS

To have a global view of each of the domain pairs, we select four sets of features to build our analysis on: CCA-related features, general dataset characteristics, descriptive statistics, and divergence features. The goal of our analysis is to understand the importance and effect scale each of these factors on cross-domain recommendations. In other words, we would like to investigate the reason behind different results that we get for each of the approaches. Is there a data characteristic that can significantly predict the results?

For the CCA-related features, we look at the number of components than can be found in the CCA analysis of the two domains. Each of these components, include an r-value and p-value that indicate the strength and significance of the canonical correlations. So, we look at the number of significant correlations between the components (with 95% confidence) and the number of components with r-value that is bigger than a threshold. To be more exact, we picked 0.8, 0.9, and 0.95 thresholds for r-values based on CCA guidelines [24]. In addition to the above, we look at the r-value for the first component (with strongest correlation), average correlations of the first five components, and average correlations of all of the discovered components.

For the single-domain and domain pair characteristics, we look at both general dataset statistics and descriptive statistics. For general dataset statistics, we look at number of users, number of items in each of the domains, density of ratings in each of the domains, and their ratios with respect to each other. For example, we look at the ratio between rating densities for domain pairs, the ratio of user numbers to source item numbers, and target item numbers to source item numbers.

For descriptive statistics of the domain pairs, we look at the rating values. As measures of central tendency, we choose average, median, and mode of rating values for both source and target domains. For dispersion measures, we look at the variance, kurtosis, and skewness of all ratings in both source and target domains.

Eventually, since we would like to measure the relationship between the rating values in source and target domains, we use divergence features. To be more specific, we look at the KL-divergence between all of the ratings in the source domain and all of the ratings in the target domain. However, since recommender systems rely on the similarity among users, we also look at the KL-divergence of ratings in the user level. To do this, we calculate the KL-divergence between each user's ratings in source and target domains. Then, we use average, median, and variance of these user-based KL-divergences to calculate the global user-based KL-divergence features for each domain pair.

Eventually, we end up with 33 different features for each domain pair. These data characteristics and their values are listed in Sections A.2, B.1, and C.1 in the Appendix sections for each of the datasets.

## 7.2 CORRELATION ANALYSIS

In this section, we seek to answer research questions Q2.1 and Q2.2. We analyze the correlation between each of the mentioned data characteristics in Section 7.1 and the error of single and cross-domain recommendations to figure out the features that lead to a fit domain pair. More specifically, we look at the correlation of single-domain data characteristics with the error of single-domain recommenders and the correlation of both single and cross-domain features with the error of cross-domain recommenders.

Additionally, since we have discovered a correlation between the single-domain and cross-domain error results in the previous chapter, we look at the relative improvement that we achieve in cross-domain recommendations, compared to the single-domain one. Thus, we define an "Improvement Ratio" factor as a dependent variable. Then, we run bivariate correlation analysis on data characteristics defined in Section 7.1 as independent variables.

The improvement ratio of algorithm $a_1$ over algorithm $a_2$ with the source domain $s_i$ and target domain $d_j$ ($\mathrm{IR}_{a_1,a_2}(s_i, d_j)$) is equivalent to the improvement of error of algorithm $a_1$ over algorithm $a_2$, normalized by error of algorithm $a_2$ in the source domain $s_i$ and target domain $d_j$ (Equation 7.1).

$$\mathrm{IR}_{a_1,a_2}(s_i, d_j) = \frac{\mathrm{Error}_{a_1}(s_i, d_j) - \mathrm{Error}_{a_2}(s_i, d_j)}{\mathrm{Error}_{a_2}(s_i, d_j)} \tag{7.1}$$

In the following sections, the correlations with p-value $< 0.05$ are shown with one star, the ones with p-value $< 0.01$ are shown with two stars, and the ones with p-value $< 0.001$ are shown with three stars. Also, we count all of the correlations with p-value $< 0.05$ as significant correlations.

### 7.2.1 Correlation Analysis for the Supermarket Purchase Dataset

**7.2.1.1 Correlation Analysis of Errors** In this section, we look at the bivariate Pearson correlation of each of the data statistics with the error of each of the algorithms. Table 17 shows these correlations with the RMSE of algorithms and Table 16 shows them with the MAE of algorithms.

As we can see, the total KL-divergence of ratings in the source and target domains, the mode of source domain rating values and the average CCA correlations between the source and target domains do not have any significant correlations with the RMSE and MAE of any of the algorithms. Also, the average and median of user-based KL-divergences of source and target domains do not have any significant correlations with the RMSE of algorithms. However, they have a negative correlation with MAE of CD-CCA. The significant correlation of these two factors with the SD-SVD error is meaningless, because in SD-SVD, we only use the target domain data.

Except for the average CCA correlations between the source and target domains, the rest of CCA-related features have at least one significant correlation with the error of algorithms. For example, the number of significant CCA correlations, is significantly correlated with RMSE of CD-CCA, RMGM, CD-SVD, and SD-SVD; the average of first five components' CCA is significantly correlated with RMSE of CMF, RMGM, CD-SVD, and SD-SVD; and the value of first component's CCA is significantly correlated with RMSE of CMF, RMGM, and SD-SVD.

Table 16: Correlations of data characteristics with MAE of algorithms on the Supermaket dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| variables | CD-CCA MAE | CMF MAE | RMGM MAE | CD-SVD MAE | SD-SVD MAE |
|---|---|---|---|---|---|
| user size | 0.3262* | -0.0441 | -0.5925*** | 0.4777*** | 0.7593*** |
| source item size | 0.4702*** | 0.0043 | -0.2012 | 0.3167* | 0.2956* |
| target item size | -0.1585 | 0.0074 | -0.5761*** | 0.2363 | 0.3918** |
| source density | -0.0326 | 0.3506* | 0.3546* | 0.5849*** | -0.1233 |
| target density | 0.039 | -0.8624*** | -0.6611*** | -0.1777 | 0.2763 |
| total KL-divergence | -0.1717 | -0.1384 | 0.0191 | -0.1786 | -0.1868 |
| mean user KL-divergence | -0.3032* | -0.0836 | 0.2179 | 0.0352 | -0.3761** |
| median user KL-divergence | -0.2994* | -0.0645 | 0.2231 | 0.1402 | -0.3543* |
| variance user KL-divergence | -0.1676 | 0.3838** | 0.8286*** | -0.0679 | -0.6455*** |
| source mean rating | 0.0049 | 0.3421* | 0.247 | 0.6268*** | 0.0283 |
| target mean rating | 0.2885* | -0.7415*** | -0.7859*** | -0.016 | 0.5452*** |
| source median rating | -0.0215 | 0.3475* | 0.2726 | 0.6211*** | 0.0025 |
| target median rating | 0.3113* | -0.7554*** | -0.8048*** | -0.0283 | 0.5476*** |
| source mode rating | 0.09 | 0.189 | -0.0884 | 0.2697 | 0.2319 |
| target mode rating | -0.0446 | -0.6463*** | -0.3259* | 0.0486 | 0.2391 |
| source var rating | 0.005 | 0.3111* | 0.1997 | 0.5877*** | 0.0417 |
| target var rating | 0.2404 | -0.6977*** | -0.6131*** | -0.0371 | 0.4268** |
| source kurtosis rating | -0.1073 | -0.1338 | -0.0356 | -0.2982* | -0.1064 |
| target kurtosis rating | -0.1827 | 0.2162 | 0.6107*** | -0.1987 | -0.4793*** |
| source skewness rating | -0.1281 | -0.1418 | -0.0408 | -0.3318* | -0.1144 |
| target skewness rating | -0.1801 | 0.2392 | 0.6676*** | -0.204 | -0.5167*** |
| user to source item ratio | -0.3496* | -0.0536 | -0.1673 | -0.0662 | 0.1397 |
| user to target item ratio | 0.431** | 0.0189 | 0.3157* | 0.0514 | 0.0189 |
| source to target item ratio | 0.5161*** | 0.024 | 0.3378* | 0.0893 | -0.0923 |
| source to target density ratio | -0.1478 | 0.4109** | 0.644*** | 0.4444** | -0.4935*** |
| CCA correlation $\geq 0.80$ | 0.0129 | -0.1914 | -0.4423** | 0.1974 | 0.2572 |
| CCA correlation $\geq 0.90$ | 0.0492 | -0.2208 | -0.5352*** | 0.2183 | 0.3403* |
| CCA correlation $\geq 0.95$ | 0.0256 | -0.2492 | -0.5717*** | 0.204 | 0.3873** |
| average correlation | -0.0914 | -0.1572 | -0.247 | 0.1455 | 0.114 |
| first component correlation | 0.1104 | -0.1576 | -0.5075*** | 0.317* | 0.3793** |
| first 5 components correlation | 0.1145 | -0.2227 | -0.5938*** | 0.2587 | 0.4096** |
| # components | 0.1424 | 0.0196 | -0.4671*** | 0.3329* | 0.4781*** |
| # significant correlations | -0.0863 | -0.016 | -0.5925*** | 0.3483* | 0.525*** |

Table 17: Correlations of data characteristics with RMSE of algorithms on the Supermaket dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| variables | CD-CCA RMSE | CMF RMSE | RMGM RMSE | CD-SVD RMSE | SD-SVD RMSE |
|---|---|---|---|---|---|
| user size | -0.0598 | -0.1703 | -0.5693*** | 0.4108** | 0.7439*** |
| source item size | 0.3265* | -0.2235 | -0.1831 | 0.2758 | 0.2827* |
| target item size | -0.3395* | -0.2111 | -0.5343*** | 0.1757 | 0.363** |
| source density | 0.1617 | 0.1172 | 0.3587* | 0.6299*** | -0.0981 |
| target density | -0.3796** | -0.4313** | -0.7199*** | -0.2782 | 0.2448 |
| total KL-divergence | -0.1587 | 0.0509 | -0.0003 | -0.2103 | -0.2006 |
| mean user KL-divergence | -0.1694 | 0.1084 | 0.1890 | 0.0332 | -0.3874** |
| median user KL-divergence | -0.1741 | 0.1612 | 0.1862 | 0.1332 | -0.359* |
| variance user KL-divergence | 0.3158* | 0.4961*** | 0.831*** | 0.0814 | -0.6189*** |
| source mean rating | 0.1384 | 0.1345 | 0.2583 | 0.6772*** | 0.0516 |
| target mean rating | -0.2404 | -0.4377** | -0.8317*** | -0.1303 | 0.5145*** |
| source median rating | 0.1270 | 0.1470 | 0.2847* | 0.6779*** | 0.0267 |
| target median rating | -0.2247 | -0.4583*** | -0.8498*** | -0.1496 | 0.5135*** |
| source mode rating | -0.0173 | -0.0247 | -0.0738 | 0.2701 | 0.2444 |
| target mode rating | -0.2688 | -0.3138* | -0.3695** | 0.0373 | 0.2477 |
| source var rating | 0.1122 | 0.1261 | 0.2051 | 0.6202*** | 0.0639 |
| target var rating | -0.2073 | -0.3184* | -0.669*** | -0.1167 | 0.4091** |
| source kurtosis rating | -0.1199 | 0.0726 | -0.0480 | -0.3341* | -0.1106 |
| target kurtosis rating | 0.0959 | 0.5192*** | 0.5725*** | -0.1194 | -0.4642*** |
| source skewness rating | -0.1420 | 0.0780 | -0.0549 | -0.3681** | -0.1178 |
| target skewness rating | 0.1299 | 0.529*** | 0.6307*** | -0.1139 | -0.4961*** |
| user to source item ratio | -0.4389** | 0.1225 | -0.1738 | -0.0686 | 0.1429 |
| user to target item ratio | 0.437** | 0.1710 | 0.2876* | 0.0810 | 0.0421 |
| source to target item ratio | 0.5717*** | 0.0210 | 0.3191* | 0.1046 | -0.0818 |
| source to target density ratio | 0.2573 | 0.2166 | 0.647*** | 0.5296*** | -0.4641*** |
| CCA correlation $\geq 0.80$ | -0.2278 | -0.3547* | -0.4285** | 0.1043 | 0.2127 |
| CCA correlation $\geq 0.90$ | -0.2289 | -0.3867** | -0.5187*** | 0.1224 | 0.2982* |
| CCA correlation $\geq 0.95$ | -0.2576 | -0.342* | -0.5585*** | 0.1185 | 0.3557* |
| average correlation | -0.2428 | -0.2628 | -0.2417 | 0.0763 | 0.0798 |
| first component correlation | -0.0936 | -0.3517* | -0.4814*** | 0.2479 | 0.361** |
| first 5 components correlation | -0.1678 | -0.3901** | -0.5733*** | 0.1678 | 0.3747** |
| # components | -0.0752 | -0.2096 | -0.4355** | 0.2862* | 0.4628*** |
| # significant correlations | -0.3511* | -0.2179 | -0.5587*** | 0.2846* | 0.5005*** |

As we expected to have better cross-domain recommendations when having a high canonical correlation between the domain pairs, these CCA-related features have negative correlations with RMSE of CD-CCA, RMGM, and CMF. It means that the RMSE of these algorithms are lower when there is a high canonical correlation between the source and target domains. However, as it is shown in the table, although CD-SVD is also a cross-domain recommender, these correlations are always positive for its error. It means that the error of CD-SVD grows more with the higher CCA between the source and target domains. Also, we can see that although SD-SVD is a single-domain algorithm (thus there should not be any meaningful correlations between its error and CCA-based features), there is a significant positive correlation between the error of SD-SVD and most of the CCA-related features. As we have seen in section 5.2, the error of algorithms, especially for CD-SVD and SD-SVD, are highly correlated in the Supermarket dataset. Consequently, we hypothesize that the positive correlation between the error of CD-SVD and the CCA-related features is because of the same factors that create a positive correlation between the error of SD-SVD and these features. Especially, because the magnitude and significance of this positive correlation is higher for RMSE and MAE of SD-SVD, compared to CD-SVD.

Among the general dataset characteristics, the density of target domain has a significant negative correlation with RMSE of CD-CCA, RMGM, and CMF and MAE of RMGM and CMF. Thus, we will have a lower error rate when there is more user rating information available in the target domain. The denser the source domain is, the higher error we will have in CD-SVD and RMGM algorithms. It means that more information in the source domain can harm more than help in these two cross-domain recommenders. One interesting observation is the correlation between number of users and RMSE of CD-SVD and SD-SVD (and MAE of CD-CCA, CD-SVD, and SD-SVD). As the number of users grow, we expect to have a better understanding of various user tastes, and thus better recommendations. However, for these two algorithms in the Supermarket dataset, this relationship works in reverse. Also, we see that as the number of users grow compared to the number of source domain items (when the user-item source domain matrix gets taller), we achieve significantly less error from CD-CCA. However, as the target domain's user-item matrix gets taller, we see an increase in error of CD-CCA and RMGM. Another general factor with a large correlation with the errors is the density ratio of source to target domains. Having a higher density source domain, compared to target domain, results in worse recommendations from RMGM, CD-SVD, and CMF; and (meaninglessly) better recommendations in SD-SVD.

Among the descriptive statistics features, most of them have a significant relationship with CMF. While the source domain's central tendency measures have a positive correlation with MAE of CMF, these features from the target domain are negatively correlated with CMF's error. The target domain central tendency features are also negatively correlated with RMGM's errors and positively correlated with SD-SVD's. For the dispersion statistics, we can see that SD-SVD performs worse when there is more variance in the target domain ratings; but the cross-domain recommenders work better in this case. This relationship is the reverse for target ratings' kurtosis and skewness. More specifically, the RMSE of RMGM and MAE of RMGM and CMF increases significantly when the target data ratings are skewed and have more kurtosis.

In general, we can see that RMGM and SD-SVD have the largest number of significant correlations with the data features. For SD-SVD, many of these correlations do not impose any meaningful relationship, because it only uses the target domains data and many of the features are calculated based on domain pairs. We can get a better understanding of these correlations by looking at the scatter plot of these features against the error of algorithms. These scatter plots can be found in Appendix A.5.

**7.2.1.2 Correlation Analysis of Improvement Ratio** In this section, we look at the correlation of data features with the improvement that can be achieved in the recommendation results by using cross-domain recommenders, instead of the single-domain recommender. Tables 19 and 18 show the improvement ratio (IR) of each of the cross-domain algorithms over SD-SVD for RMSE and MAE of the results.

Table 18: Correlations of data characteristics with MAE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Supermarket dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001

| variables | IR for CD-CCA over SD-SVD for all the pairs | IR for CMF over SD-SVD for all the pairs | IR for RMGM over SD-SVD for all the pairs | IR for CD-SVD over SD-SVD for all the pairs |
|---|---|---|---|---|
| user size | 0.6803*** | 0.5659*** | 0.6556*** | 0.3639** |
| source item size | 0.121 | 0.2002 | 0.2251 | 0.0111 |
| target item size | 0.4528*** | 0.3081* | 0.5077*** | 0.2047 |
| source density | -0.215 | -0.3327* | -0.3177* | -0.6189*** |
| target density | 0.3528* | 0.7081*** | 0.535*** | 0.3865** |
| total KL-divergence | -0.0982 | -0.017 | -0.0588 | -0.0517 |
| mean user KL-divergence | -0.3103* | -0.2168 | -0.2866* | -0.4413** |
| median user KL-divergence | -0.2831* | -0.2176 | -0.2859* | -0.522*** |
| variance user KL-divergence | -0.7482*** | -0.7879*** | -0.8332*** | -0.5933*** |
| source mean rating | -0.0832 | -0.2316 | -0.2007 | -0.4955*** |
| target mean rating | 0.5283*** | 0.8181*** | 0.7054*** | 0.5032*** |
| source median rating | -0.1058 | -0.258 | -0.2296 | -0.5165*** |
| target median rating | 0.5376*** | 0.8387*** | 0.7278*** | 0.5208*** |
| source mode rating | 0.1709 | 0.0541 | 0.118 | 0.013 |
| target mode rating | 0.2383 | 0.439** | 0.2402 | 0.1518 |
| source var rating | -0.043 | -0.1812 | -0.1501 | -0.4492** |
| target var rating | 0.3833** | 0.6712*** | 0.5252*** | 0.3928** |
| source kurtosis rating | 0.0038 | 0.0595 | 0.0318 | 0.1437 |
| target kurtosis rating | -0.484*** | -0.5313*** | -0.6041*** | -0.2927* |
| source skewness rating | 0.0105 | 0.063 | 0.036 | 0.1657 |
| target skewness rating | -0.5314*** | -0.5777*** | -0.6583*** | -0.3288* |
| user to source item ratio | 0.302* | 0.1475 | 0.1754 | 0.218 |
| user to target item ratio | -0.0984 | -0.018 | -0.1744 | -0.0164 |
| source to target item ratio | -0.2726 | -0.099 | -0.2426 | -0.1756 |
| source to target density ratio | -0.5963*** | -0.6616*** | -0.6594*** | -0.9057*** |
| CCA correlation $\geq 0.80$ | 0.2497 | 0.324* | 0.3699** | 0.0698 |
| CCA correlation $\geq 0.90$ | 0.3196* | 0.3911** | 0.4523*** | 0.1325 |
| CCA correlation $\geq 0.95$ | 0.3827** | 0.4325** | 0.4914*** | 0.1998 |
| average correlation | 0.1421 | 0.2046 | 0.202 | -0.0193 |
| first component correlation | 0.3683** | 0.4052** | 0.4787*** | 0.1259 |
| first 5 components correlation | 0.3779** | 0.4424** | 0.5208*** | 0.1759 |
| # components | 0.4109** | 0.3187* | 0.4474** | 0.1934 |
| # significant correlations | 0.536*** | 0.383** | 0.5435*** | 0.2307 |

Table 19: Correlations of data characteristics with RMSE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Supermarket dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| variables | IR for CD-CCA over SD-SVD for all the pairs | IR for CMF over SD-SVD for all the pairs | IR for RMGM over SD-SVD for all the pairs | IR for CD-SVD over SD-SVD for all the pairs |
|---|---|---|---|---|
| user size | 0.6831*** | 0.3683** | 0.643*** | 0.3944** |
| source item size | 0.1191 | 0.2644 | 0.2109 | 0.0383 |
| target item size | 0.4528*** | 0.3163* | 0.5048*** | 0.226 |
| source density | -0.2177 | -0.1307 | -0.3227* | -0.617*** |
| target density | 0.4098** | 0.4021** | 0.5955*** | 0.4435** |
| total KL-divergence | -0.0935 | -0.0932 | -0.0487 | -0.0337 |
| mean user KL-divergence | -0.303* | -0.2128 | -0.2708 | -0.4325** |
| median user KL-divergence | -0.2719 | -0.2554 | -0.2602 | -0.4964*** |
| variance user KL-divergence | -0.7727*** | -0.6405*** | -0.8541*** | -0.6848*** |
| source mean rating | -0.0875 | -0.0993 | -0.2099 | -0.5058*** |
| target mean rating | 0.5881*** | 0.5019*** | 0.7579*** | 0.5738*** |
| source median rating | -0.1096 | -0.1202 | -0.2395 | -0.5312*** |
| target median rating | 0.5926*** | 0.5187*** | 0.7777*** | 0.5925*** |
| source mode rating | 0.197 | 0.0901 | 0.1177 | 0.0236 |
| target mode rating | 0.2921* | 0.2629 | 0.289* | 0.1763 |
| source var rating | -0.0466 | -0.0782 | -0.1543 | -0.4456** |
| target var rating | 0.4506** | 0.3617** | 0.5849*** | 0.4497** |
| source kurtosis rating | 0.0033 | -0.0682 | 0.0375 | 0.1682 |
| target kurtosis rating | -0.5066*** | -0.6336*** | -0.6036*** | -0.3583* |
| source skewness rating | 0.0105 | -0.0738 | 0.0435 | 0.1904 |
| target skewness rating | -0.5515*** | -0.6488*** | -0.6574*** | -0.3959** |
| user to source item ratio | 0.308* | -0.0476 | 0.1849 | 0.2117 |
| user to target item ratio | -0.1169 | -0.1525 | -0.1889 | -0.0277 |
| source to target item ratio | -0.2905* | -0.0587 | -0.2606 | -0.1778 |
| source to target density ratio | -0.5986*** | -0.3255* | -0.6601*** | -0.915*** |
| CCA correlation $\geq$ 0.80 | 0.2691 | 0.3901** | 0.3682** | 0.1109 |
| CCA correlation $\geq$ 0.90 | 0.3447* | 0.4288** | 0.4564*** | 0.179 |
| CCA correlation $\geq$ 0.95 | 0.4095** | 0.3914** | 0.5029*** | 0.2453 |
| average correlation | 0.1557 | 0.2835* | 0.1968 | 0.0093 |
| first component correlation | 0.3721** | 0.4353** | 0.4743*** | 0.1675 |
| first 5 components correlation | 0.3978** | 0.454*** | 0.5256*** | 0.2233 |
| # components | 0.4188** | 0.3029* | 0.4439** | 0.2178 |
| # significant correlations | 0.5518*** | 0.3387* | 0.5442*** | 0.2594 |

As shown in the tables, number of shared users, target domain's density, and average, median, and variance of user ratings in the target domain are the factors that are positively correlated with the improvement ratios of all cross-domain recommenders versus SD-SVD, calculated on either RMSE or MAE. Variance in KL-divergence of user ratings between source and target domains, kurtosis and skewness of target domain ratings, and the density ratio of source domain to target domain, all have a negative correlation with the improvement ratios of all algorithms.

Also, we can see that the number of items in the source domain, the KL-divergence of all ratings in the source and target domains, mode, kurtosis, and skewness of rating values in the source domain, the ratio of number of common users to number of target domain items, and the ratio between number of items in the source and target domains do not have any significant correlations with the IRs.

Among the CCA-related features, almost all of them (except average canonical correlation) are positively and significantly correlated with IR of CD-CCA, RMGM, and CMF. Interestingly, the improvement of CD-SVD over SD-SVD does not follow a similar rule: the correlations are weak and not significant.

The scatter plot of these features against the improvement ratio of algorithms can be found in Appendix A.5.

### 7.2.2 Correlation Analysis for Yelp Dataset

**7.2.2.1 Correlation Analysis of Errors** Tables 21 and 20 show the bivariate Pearson correlation between the RMSE and MAE of algorithms and the dataset characteristics for the Yelp dataset.

Table 20: Correlations of data characteristics with MAE of algorithms on the Yelp dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| variables | CD-CCA MAE | CMF MAE | RMGM MAE | CD-SVD MAE | SD-SVD MAE |
|---|---|---|---|---|---|
| user size | -0.1747* | -0.1206 | -0.4416*** | 0.0506 | 0.0806 |
| source item size | -0.1374 | -0.135 | -0.2912*** | 0.0178 | 0.0176 |
| target item size | -0.1466 | -0.096 | -0.545*** | 0.0232 | 0.0228 |
| source density | 0.16* | 0.3345*** | 0.3647*** | -0.0974 | -0.0949 |
| target density | 0.0822 | 0.2441** | 0.2735*** | -0.1189 | -0.0884 |
| total KL-divergence | 0.2823*** | 0.5248*** | 0.1583* | -0.0224 | 0.051 |
| mean user KL-divergence | -0.1564 | 0.0286 | -0.2179** | -0.0089 | -0.0217 |
| median user KL-divergence | -0.1174 | -0.0106 | -0.2637*** | -0.0159 | -0.0035 |
| variance user KL-divergence | 0.1848* | 0.2528** | 0.5671*** | -0.0207 | -0.0797 |
| source mean rating | -0.2491** | 0.01 | 0.0737 | -0.2444** | -0.243** |
| target mean rating | -0.3789*** | -0.1638* | 0.4904*** | -0.3583*** | -0.457*** |
| source median rating | 0.0036 | 0.0862 | 0.0291 | -0.1302 | -0.097 |
| target median rating | 0.0339 | 0.0495 | 0.4978*** | -0.2154** | -0.2944*** |
| source mode rating | 0.0754 | 0.1185 | 0.0139 | -0.0616 | -0.0168 |
| target mode rating | 0.4441*** | 0.2908*** | 0.7259*** | 0.0229 | -0.0184 |
| source var rating | 0.364*** | 0.2485** | -0.044 | 0.1227 | 0.2128** |
| target var rating | 0.9222*** | 0.4855*** | 0.4326*** | 0.301*** | 0.33*** |
| source kurtosis rating | -0.2328** | -0.0632 | 0.0959 | -0.2103** | -0.2414** |
| target kurtosis rating | -0.5195*** | -0.2627*** | 0.2639*** | -0.3639*** | -0.4431*** |
| source skewness rating | 0.1836* | -0.0448 | -0.0551 | 0.2233** | 0.219** |
| target skewness rating | 0.2057* | 0.0704 | -0.5734*** | 0.3029*** | 0.3966*** |
| user to source item ratio | -0.222** | -0.1113 | -0.4632*** | 0.0507 | 0.0868 |
| user to target item ratio | -0.2191** | -0.2359** | -0.126 | 0.0498 | 0.0815 |
| source to target item ratio | -0.1125 | -0.2127** | 0.1437 | -0.043 | -0.0586 |
| source to target density ratio | 0.3044*** | 0.1737* | 0.3143*** | 0.1294 | 0.0595 |
| CCA correlation $\geq 0.80$ | -0.0154 | -0.082 | -0.2807*** | -0.0238 | -0.01 |
| CCA correlation $\geq 0.90$ | -0.0873 | -0.1467 | -0.332*** | -0.0243 | -0.0212 |
| CCA correlation $\geq 0.95$ | -0.0637 | -0.0761 | -0.1228 | -0.1103 | -0.0856 |
| average correlation | -0.0236 | -0.1022 | -0.3981*** | 0.0594 | 0.0884 |
| first component correlation | -0.0927 | -0.1681* | -0.4294*** | 0.1428 | 0.1151 |
| first 5 components correlation | -0.1884* | -0.4392*** | -0.5319*** | 0.1718* | 0.1313 |
| # components | -0.1727* | -0.1412 | -0.5009*** | 0.0548 | 0.0736 |
| # significant correlations | -0.1748* | -0.1282 | -0.5285*** | 0.0487 | 0.0659 |

Table 21: Correlations of data characteristics with RMSE of algorithms on the Yelp dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| variables | CD-CCA RMSE | CMF RMSE | RMGM RMSE | CD-SVD RMSE | SD-SVD RMSE |
|---|---|---|---|---|---|
| user size | -0.1782* | 0.0039 | -0.4475*** | -0.1745* | -0.1455 |
| source item size | -0.1239 | 0.0087 | -0.2754*** | -0.1274 | -0.0953 |
| target item size | -0.125 | -0.0537 | -0.5613*** | -0.1445 | -0.1225 |
| source density | 0.0515 | 0.0407 | 0.3396*** | -0.1161 | -0.1201 |
| target density | -0.0502 | -0.0098 | 0.2513** | -0.1346 | -0.1525 |
| total KL-divergence | 0.194* | 0.1772* | 0.1376 | -0.0669 | -0.2068** |
| mean user KL-divergence | -0.1597* | -0.0001 | -0.2284** | -0.0532 | -0.1093 |
| median user KL-divergence | -0.1532 | -0.0104 | -0.2754*** | -0.0821 | -0.142 |
| variance user KL-divergence | 0.2526** | 0.1367 | 0.5741*** | 0.1234 | 0.0944 |
| source mean rating | -0.2173** | -0.0224 | 0.0719 | -0.0759 | -0.1682* |
| target mean rating | -0.2679*** | -0.2141** | 0.5041*** | -0.4038*** | -0.33*** |
| source median rating | -0.0075 | -0.0409 | 0.0233 | 0.0789 | 0.0078 |
| target median rating | 0.1271 | -0.0604 | 0.4941*** | -0.1213 | -0.0491 |
| source mode rating | 0.0346 | -0.0233 | 0.0045 | 0.0972 | 0.0156 |
| target mode rating | 0.495*** | 0.0652 | 0.7302*** | 0.3559*** | 0.3973*** |
| source var rating | 0.2939*** | 0.0614 | -0.0519 | 0.1339 | 0.1176 |
| target var rating | 0.876*** | 0.2667*** | 0.4259*** | 0.6789*** | 0.6415*** |
| source kurtosis rating | -0.2009* | -0.0512 | 0.0958 | -0.1039 | -0.1651* |
| target kurtosis rating | -0.4062*** | -0.2154** | 0.2753*** | -0.5027*** | -0.4313*** |
| source skewness rating | 0.1637* | 0.0239 | -0.0514 | 0.0912 | 0.174* |
| target skewness rating | 0.0742 | 0.1252 | -0.5846*** | 0.2747*** | 0.2027* |
| user to source item ratio | -0.2427** | -0.0581 | -0.4812*** | -0.2047** | -0.1831* |
| user to target item ratio | -0.2532** | 0.0121 | -0.1159 | -0.1595* | -0.1282 |
| source to target item ratio | -0.1308 | -0.0265 | 0.1572 | -0.0406 | -0.0225 |
| source to target density ratio | 0.3824*** | 0.0793 | 0.3196*** | 0.2477** | 0.2871*** |
| CCA correlation $\geq 0.80$ | -0.0765 | -0.0945 | -0.296*** | -0.1704* | -0.1401 |
| CCA correlation $\geq 0.90$ | -0.127 | -0.1269 | -0.3452*** | -0.2301** | -0.207** |
| CCA correlation $\geq 0.95$ | -0.102 | -0.0678 | -0.1409 | -0.2932*** | -0.3426*** |
| average correlation | -0.095 | -0.0967 | -0.4118*** | -0.0909 | -0.0798 |
| first component correlation | -0.072 | -0.099 | -0.4246*** | -0.1498 | -0.1036 |
| first 5 components correlation | -0.0933 | -0.133 | -0.5075*** | 0.045 | 0.1213 |
| # components | -0.1536 | -0.0353 | -0.4974*** | -0.1514 | -0.1205 |
| # significant correlations | -0.1601* | -0.0375 | -0.5343*** | -0.1606* | -0.1335 |

As we can see in these tables, median and mode of source domain ratings do not have any significant correlations with any of the errors of the algorithms. Average of ratings in the target domain, kurtosis of target domain ratings, and variance in target domain ratings, all have significant correlations with the error of all algorithms. The more variance we see in the target ratings, the more the RMSE and MAE of all algorithms will be. However average of ratings in the target domain and kurtosis of target domain ratings have a tricky correlation with error of algorithms. While they are both negatively correlated with RMSE and MAE of CD-CCA, CD-SVD, CMF, and SD-SVD, their correlation with RMGM's MAE and RMSE is positive.

Also, source domain's density, KL-divergence between all ratings of the two domains, variance in user-based KL-divergences between the domains, mode of the rating values in the target domain, and density ratio between source and target domains are all positively and significantly correlated with either MAE or RMSE of CD-CCA, CMF, and RMGM algorithms.

The number of users has a significant negative correlation with RMSE of all cross-domain algorithms except CMF, and with MAE of CD-CCA and RMGM. This means that as the number of common users between the domains grow, we achieve better cross-domain recommendations in these algorithms. The same relationship exists between the ratio of user numbers to number of items in the source domain. In this case, we have a significant negative relationship between this factor and RMSE of CD-CCA, CD-SVD, and RMGM and with MAE of CD-CCA and RMGM.

Among the CCA-related features, they mostly have a negative correlation with the error of algorithms. But, we can see more significant correlations with MAE of algorithms. For example, number of significantly correlated components, maximum number of components, average correlation of first five components all have a negative significant correlation with MAE of CD-CCA and RMGM. But, only number of significantly correlated components is correlated with RMSE of CD-CCA. Also, only MAE of CMF is significantly correlated with the value of first component's correlation and average correlation of first five components. For CD-SVD, the correlations are confusing: while the RMSE of this algorithm has a negative significant correlation with the number of significantly correlated components, its MAE is positively correlated with the average correlation of first five components. Again, this phenomenon can be because of the high correlation of error in SD-SVD and CD-SVD, and the fact that SD-SVD has a positive (although non-significant) correlation with the average correlation of first five components.

Again, we can see that RMGM has the most number of significant correlations with the data features.

We can look at the scatter plot of these features against the error of algorithms in Appendix B.4 to get a better understanding of these correlations. The blue circles show the non-significant improvement ratios and the red crosses show the significant ones. In these pictures, you can see the improvement ratio of all of the algorithms compared to each other (not only the improvement ratio of cross-domain algorithms over SD-SVD).

**7.2.2.2 Correlation Analysis of Improvement Ratio** If we look at the IR of cross-domain algorithms vs SD-SVD in Tables 23 (for their RMSE) and 22 (for their MAE), we can see that most of the CCA-related features have a positive correlation with the IRs. More specifically, the average correlation of the first component is positively and significantly correlated with the improvement of all cross-domain RMSEs, over the single-domain one; and is positively and significantly correlated with the IR of CD-CCA, CMF, and RMGM, calculated over MAEs. Also, the maximum number of components, and number of components with significant correlations are positively correlated with IR of CD-CCA, CMF, and RMGM, calculated over MAEs. The complicated case here is the negative significant correlation of the number of components with more than 0.95 CCA, with MAE-based IR of CD-CCA, and RMSE-based IR of CD-CCA and CD-SVD. One of the possible reasons for this relationship can be the few number of cases in which there exists a CCA more than 0.95 between the components. In other words, since very few of the discovered components, in a few of domain pairs, have such a high correlation, this result can be less reliable.

Table 22: Correlations of data characteristics with MAE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Yelp dataset; *: significant with p-value $<$ 0.05; **: significant with p-value $<$ 0.01; ***: significant with p-value $<$ 0.001

| variables | IR for CD-CCA over SD-SVD for all the pairs | IR for CMF over SD-SVD for all the pairs | IR for RMGM over SD-SVD for all the pairs | IR for CD-SVD over SD-SVD for all the pairs |
|---|---|---|---|---|
| user size | 0.1945* | 0.1661* | 0.3324*** | 0.0601 |
| source item size | 0.1296 | 0.1556 | 0.2238** | 0.0034 |
| target item size | 0.1401 | 0.1168 | 0.3949*** | -0.0094 |
| source density | -0.3325*** | -0.4406*** | -0.3669*** | -0.0095 |
| target density | -0.3123*** | -0.3833*** | -0.3357*** | 0.0706 |
| total KL-divergence | -0.189* | -0.4209*** | -0.0924 | 0.159* |
| mean user KL-divergence | 0.1309 | -0.0198 | 0.2031* | -0.0284 |
| median user KL-divergence | 0.1055 | 0.0086 | 0.2119** | 0.0371 |
| variance user KL-divergence | -0.2122** | -0.2822*** | -0.3666*** | -0.119 |
| source mean rating | -0.1373 | -0.1682* | -0.2336** | 0.1113 |
| target mean rating | -0.4017*** | -0.2806*** | -0.7135*** | -0.129 |
| source median rating | -0.1324 | -0.112 | -0.1177 | 0.1138 |
| target median rating | -0.5065*** | -0.375*** | -0.6256*** | -0.1723* |
| source mode rating | -0.0997 | -0.1452 | -0.0552 | 0.1232 |
| target mode rating | -0.36*** | -0.3053*** | -0.5133*** | -0.1034 |
| source var rating | -0.0528 | -0.1068 | 0.1 | 0.1327 |
| target var rating | -0.2239** | -0.1909* | -0.0559 | -0.0217 |
| source kurtosis rating | -0.1481 | -0.1134 | -0.2662*** | 0.0278 |
| target kurtosis rating | -0.317*** | -0.1854* | -0.6055*** | -0.0963 |
| source skewness rating | 0.1776* | 0.1997* | 0.2403** | -0.1069 |
| target skewness rating | 0.4771*** | 0.3291*** | 0.7602*** | 0.1421 |
| user to source item ratio | 0.2423** | 0.1617* | 0.3449*** | 0.0689 |
| user to target item ratio | 0.239** | 0.2863*** | 0.1233 | 0.0709 |
| source to target item ratio | 0.0394 | 0.1883* | -0.1107 | -0.0257 |
| source to target density ratio | -0.1227 | -0.1094 | -0.1391 | -0.1999* |
| CCA correlation $\geq$ 0.80 | -0.0671 | 0.032 | 0.0993 | 0.0172 |
| CCA correlation $\geq$ 0.90 | -0.0367 | 0.0632 | 0.1282 | -0.0098 |
| CCA correlation $\geq$ 0.95 | -0.241** | -0.1169 | -0.1367 | 0.0851 |
| average correlation | 0.1189 | 0.161* | 0.3063*** | 0.0266 |
| first component correlation | 0.1327 | 0.1989* | 0.2896*** | -0.1191 |
| first 5 components correlation | 0.381*** | 0.5376*** | 0.498*** | -0.126 |
| # components | 0.2096** | 0.1952* | 0.3896*** | 0.0428 |
| # significant correlations | 0.1978* | 0.1739* | 0.399*** | 0.0346 |

Table 23: Correlations of data characteristics with RMSE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Yelp dataset; *: significant with p-value $<$ 0.05; **: significant with p-value $<$ 0.01; ***: significant with p-value $<$ 0.001

| variables | IR for CD-CCA over SD-SVD for all the pairs | IR for CMF over SD-SVD for all the pairs | IR for RMGM over SD-SVD for all the pairs | IR for CD-SVD over SD-SVD for all the pairs |
|---|---|---|---|---|
| user size | 0.0595 | -0.0385 | 0.311*** | 0.1016 |
| source item size | 0.0631 | -0.0319 | 0.2036* | 0.1186 |
| target item size | 0.0354 | 0.0369 | 0.424*** | 0.0863 |
| source density | -0.3493*** | -0.0915 | -0.4647*** | -0.1772* |
| target density | -0.3357*** | -0.0592 | -0.4544*** | -0.217** |
| total KL-divergence | -0.5609*** | -0.2323** | -0.3149*** | -0.6312*** |
| mean user KL-divergence | -0.0185 | -0.0238 | 0.1232 | -0.2226** |
| median user KL-divergence | -0.0554 | -0.023 | 0.1279 | -0.2665*** |
| variance user KL-divergence | -0.1671* | -0.1221 | -0.3997*** | -0.1324 |
| source mean rating | -0.0053 | -0.0076 | -0.184* | -0.22** |
| target mean rating | -0.005 | 0.1641* | -0.574*** | 0.2133** |
| source median rating | -0.0048 | 0.0563 | -0.0455 | -0.1801* |
| target median rating | -0.1881* | 0.0427 | -0.463*** | 0.1949* |
| source mode rating | -0.1121 | 0.0147 | -0.072 | -0.2354** |
| target mode rating | -0.0403 | 0.0381 | -0.362*** | 0.1306 |
| source var rating | -0.1785* | -0.0305 | 0.068 | -0.1209 |
| target var rating | -0.2936*** | -0.1553 | -0.0563 | -0.1369 |
| source kurtosis rating | -0.0046 | 0.0175 | -0.2067** | -0.1269 |
| target kurtosis rating | 0.0205 | 0.1409 | -0.4507*** | 0.224** |
| source skewness rating | 0.0617 | 0.006 | 0.1857* | 0.2014* |
| target skewness rating | 0.0889 | -0.0938 | 0.5841*** | -0.2213** |
| user to source item ratio | 0.0878 | 0.0267 | 0.3163*** | 0.0779 |
| user to target item ratio | 0.1547 | -0.0508 | 0.0483 | 0.1126 |
| source to target item ratio | 0.1217 | 0.0111 | -0.1276 | 0.0652 |
| source to target density ratio | -0.0688 | -0.022 | -0.0774 | 0.1345 |
| CCA correlation $\geq$ 0.80 | -0.0941 | 0.0607 | 0.1172 | 0.055 |
| CCA correlation $\geq$ 0.90 | -0.0096 | 0.0974 | 0.1758* | 0.013 |
| CCA correlation $\geq$ 0.95 | -0.1933* | 0.0035 | -0.1139 | -0.2738*** |
| average correlation | -0.0096 | 0.078 | 0.2718*** | 0.0044 |
| first component correlation | 0.1087 | 0.1094 | 0.3601*** | 0.1494 |
| first 5 components correlation | 0.422*** | 0.1877* | 0.6293*** | 0.3814*** |
| # components | 0.0739 | 0.0157 | 0.3776*** | 0.1192 |
| # significant correlations | 0.0618 | 0.0148 | 0.3966*** | 0.1035 |

Other important factors are the KL-divergence of all ratings between the source and target domains and the median ratings in the target domain. The first one is significantly and negatively correlated with the RMSE-based IR of all algorithms and the second one is significantly and negatively correlated with the MAE-based IR of all of them. This means that as the median of ratings in the target domain and the divergence between the source and target ratings grow, we will achieve less improvement when using cross-domain recommender systems.

Source domain's density, target domain's density, average, median, and variance of user-based KL-Divergence between the domains, average, median, mode, variance, and kurtosis of ratings in the source and target domains, all have a mostly negative correlation with the improvement ratio of algorithms. So, the higher they are, the less improvement we achieve in the cross-domain algorithms.

On the other hand, skewness of rating in the source and target domains, the ratio of number of users to source items, and number of users to target items have a generally positive effect on the IR of cross-domain algorithms in the Yelp dataset.

The scatter plot of these features against the improvement ratio of algorithms can be found in Appendix B.4.

### 7.2.3 Correlation Analysis for Imhonet Dataset

**7.2.3.1 Correlation Analysis of Errors** This section analyzes correlations between the Imhonet dataset features and error of algorithms. Table 25 shows these correlations for the RMSE of CD-CCA, SD-SVD, and CD-SVD; and Table 24 shows these correlations for the MAE of these algorithms. Note that we do not see any results for the number of components (# components) or number of significant correlations (# significant correlations) in these two tables. For the number of components, the reason is that the best number of discovered components for all of the domain pairs in this dataset is the same (5 components for all) and there is no variance in this feature. So, we cannot examine its correlation with the error of algorithms.

Table 24: Correlations of data characteristics with MAE of algorithms on the Imhonet dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| variables | CD-CCA MAE | CD-SVD MAE | SD-SVD MAE |
|---|---|---|---|
| user size | -0.0632 | -0.5345 | -0.403 |
| source item size | 0.604* | 0.0116 | -0.309 |
| target item size | -0.7093** | -0.3742 | -0.0922 |
| source density | -0.3402 | 0.0543 | 0.0789 |
| target density | 0.4529 | 0.4491 | 0.5155 |
| total KL-divergence | 0.5608 | 0.7411** | 0.1675 |
| mean user KL-divergence | -0.057 | -0.6151* | -0.4542 |
| median user KL-divergence | -0.0134 | -0.5658 | -0.3986 |
| variance user KL-divergence | -0.0932 | -0.6299* | -0.4812 |
| source mean rating | 0.5755 | 0.2336 | 0.1661 |
| target mean rating | -0.5898* | -0.3778 | -0.5119 |
| source mode rating | -0.0946 | 0.2031 | 0.6077* |
| target mode rating | 0.0394 | 0.1512 | -0.0998 |
| source var rating | -0.8311*** | -0.3703 | 0.0785 |
| target var rating | 0.8367*** | 0.2114 | -0.0575 |
| source kurtosis rating | 0.7129** | 0.3331 | 0.0619 |
| target kurtosis rating | -0.7564** | -0.292 | -0.2778 |
| source skewness rating | -0.5813* | -0.2654 | -0.1477 |
| target skewness rating | 0.6097* | 0.3471 | 0.4562 |
| user to source item ratio | -0.8191** | -0.033 | 0.399 |
| user to target item ratio | 0.7995** | 0.4653 | -0.0494 |
| source to target item ratio | 0.6352* | 0.642* | 0.2179 |
| source to target density ratio | -0.7088** | -0.3823 | -0.3747 |
| CCA correlation $\geq 0.80$ | 0.2312 | 0.0329 | -0.1487 |
| CCA correlation $\geq 0.90$ | -0.0181 | 0.1138 | 0.1365 |
| CCA correlation $\geq 0.95$ | -0.1191 | 0.1593 | 0.1647 |
| average correlation | 0.0332 | 0.1133 | 0.0519 |
| first component correlation | 0.0755 | 0.1857 | 0.1183 |
| first 5 components correlation | 0.0332 | 0.1133 | 0.0519 |

Table 25: Correlations of data characteristics with RMSE of algorithms on the Imhonet dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| variables | CD-CCA RMSE | CD-SVD RMSE | SD-SVD RMSE |
|---|---|---|---|
| user size | 0.0885 | -0.5475 | -0.4316 |
| source item size | 0.7073* | -0.0013 | -0.3277 |
| target item size | -0.532 | -0.3435 | -0.057 |
| source density | -0.4844 | 0.0707 | 0.0819 |
| target density | 0.1955 | 0.366 | 0.4448 |
| total KL-divergence | 0.5748 | 0.7663** | 0.2021 |
| mean user KL-divergence | 0.0451 | -0.6396* | -0.4979 |
| median user KL-divergence | 0.0736 | -0.5957* | -0.4485 |
| variance user KL-divergence | 0.0174 | -0.6483* | -0.5174 |
| source mean rating | 0.5961* | 0.1956 | 0.147 |
| target mean rating | -0.3715 | -0.2714 | -0.4159 |
| source mode rating | -0.095 | 0.1622 | 0.6* |
| target mode rating | 0.0825 | 0.2101 | -0.0658 |
| source var rating | -0.831*** | -0.3596 | 0.0908 |
| target var rating | 0.7* | 0.1671 | -0.1159 |
| source kurtosis rating | 0.7233** | 0.3059 | 0.0542 |
| target kurtosis rating | -0.567 | -0.19 | -0.178 |
| source skewness rating | -0.5825* | -0.2287 | -0.135 |
| target skewness rating | 0.4049 | 0.2307 | 0.351 |
| user to source item ratio | -0.7606** | -0.0134 | 0.4371 |
| user to target item ratio | 0.7385** | 0.4508 | -0.0675 |
| source to target item ratio | 0.5286 | 0.6142* | 0.2114 |
| source to target density ratio | -0.5769* | -0.3018 | -0.3047 |
| CCA correlation $\geq 0.80$ | 0.1209 | 0.0193 | -0.161 |
| CCA correlation $\geq 0.90$ | -0.2526 | 0.0954 | 0.1204 |
| CCA correlation $\geq 0.95$ | -0.3158 | 0.1562 | 0.1647 |
| average correlation | -0.1695 | 0.0967 | 0.0372 |
| first component correlation | -0.1128 | 0.153 | 0.0912 |
| first 5 components correlation | -0.1695 | 0.0967 | 0.0372 |

For the number of significant components, the reason is that we did not have any significant canonical correlation in any of the 5 components in any of the domains. In other words, the significance of the correlations between domain pairs, although high in value in some pairs, did not pass our threshold for the significance test.

As we can see in the tables, the significant factors that effect the error of each of these algorithms are different with each other. In CD-CCA, we can see that the number of items in the source domain, variance of ratings in the target domain, kurtosis of ratings in the source domain, and the ratio of number of users to the number of items in the target domain have a positive correlation with the errors. On the other hand, the variance in the source domain ratings, kurtosis of target domain ratings, skewness of ratings in the source domain, the ratio of number of users to number of items in the source domain, and the density ratio between source and target domains, all have a significantly negative relationship with CD-CCA's errors.

For CD-SVD, the ratio of number of users to number of items in the source domain and the KL-divergence between all ratings in the source and target domains are associated with higher errors. While more average and variance in user-based KL-divergence results in less errors.

In SD-SVD, the only significant factor is mode of ratings in the source domain, which has a positive correlation with RMSE and MAE.

Interestingly, none of the CCA-related features are significantly correlated with the error of cross-domain algorithms in this dataset. The scatter plot of these features against the error of algorithms can be found in Appendix C.4.

**7.2.3.2 Correlation Analysis of Improvement Ratio** Tables 27 and 26 show the correlation analysis between the data features and the improvement ratios of errors for CD-CCA and CD-SVD versus SD-SVD. For the reason explained in the previous section, we do not have the # components and # significant correlations features in these tables.

As we can see in these tables, the only factor that is significantly and positively correlated with the IR of CD-CCA vs. SD-SVD, is the ratio between number of users and number of items in the source domain. This means that the taller the user-item matrix of source domain, the more improvement we achieve in CD-CCA compared to SD-SVD. Also, the MAE-based improvement ratio of CD-CCA is negatively correlated with the number of items in the source domain.

Table 26: Correlations of data characteristics with MAE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Imhonet dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| variables | IR for CD-CCA over SD-SVD for all the pairs | IR for CD-SVD over SD-SVD for all the pairs |
|---|---|---|
| user size | -0.331 | 0.303 |
| source item size | -0.5731 | -0.3944 |
| target item size | 0.274 | 0.4436 |
| source density | 0.2783 | 0.034 |
| target density | 0.188 | -0.0132 |
| total KL-divergence | -0.1248 | -0.9132*** |
| mean user KL-divergence | -0.3669 | 0.3802 |
| median user KL-divergence | -0.3343 | 0.374 |
| variance user KL-divergence | -0.3768 | 0.3688 |
| source mean rating | -0.1653 | -0.1562 |
| target mean rating | -0.1171 | -0.0925 |
| source mode rating | 0.4748 | 0.3861 |
| target mode rating | -0.0746 | -0.3676 |
| source var rating | 0.4622 | 0.6446* |
| target var rating | -0.4506 | -0.3652 |
| source kurtosis rating | -0.3204 | -0.4196 |
| target kurtosis rating | 0.1794 | 0.0752 |
| source skewness rating | 0.1896 | 0.2176 |
| target skewness rating | 0.0422 | 0.0703 |
| user to source item ratio | 0.7264** | 0.5199 |
| user to target item ratio | -0.4214 | -0.7369** |
| source to target item ratio | -0.1182 | -0.6415* |
| source to target density ratio | 0.0792 | 0.1243 |
| CCA correlation $\geq 0.80$ | -0.2109 | -0.1882 |
| CCA correlation $\geq 0.90$ | 0.169 | 0.0658 |
| CCA correlation $\geq 0.95$ | 0.2491 | 0.0253 |
| average correlation | 0.0741 | -0.039 |
| first component correlation | 0.099 | -0.0593 |
| first 5 components correlation | 0.0741 | -0.039 |

Table 27: Correlations of data characteristics with RMSE-based improvement ratio of cross-domain algorithms, over SD-SVD, on the Imhonet dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| variables | IR for CD-CCA over SD-SVD for all the pairs | IR for CD-SVD over SD-SVD for all the pairs |
|---|---|---|
| user size | -0.4189 | 0.3242 |
| source item size | -0.6069* | -0.3661 |
| target item size | 0.1848 | 0.4466 |
| source density | 0.3337 | -0.0023 |
| target density | 0.2625 | -0.0251 |
| total KL-divergence | -0.0878 | -0.9198*** |
| mean user KL-divergence | -0.4437 | 0.4006 |
| median user KL-divergence | -0.4108 | 0.3908 |
| variance user KL-divergence | -0.4505 | 0.3912 |
| source mean rating | -0.1765 | -0.131 |
| target mean rating | -0.1495 | -0.084 |
| source mode rating | 0.4579 | 0.3911 |
| target mode rating | -0.0641 | -0.3997 |
| source var rating | 0.4437 | 0.6337* |
| target var rating | -0.4008 | -0.3669 |
| source kurtosis rating | -0.3068 | -0.3889 |
| target kurtosis rating | 0.1419 | 0.071 |
| source skewness rating | 0.1828 | 0.1864 |
| target skewness rating | 0.0684 | 0.0706 |
| user to source item ratio | 0.6851* | 0.5026 |
| user to target item ratio | -0.3778 | -0.7345** |
| source to target item ratio | -0.0623 | -0.6355* |
| source to target density ratio | 0.0469 | 0.1156 |
| CCA correlation $\geq$ 0.80 | -0.1567 | -0.1763 |
| CCA correlation $\geq$ 0.90 | 0.2551 | 0.0469 |
| CCA correlation $\geq$ 0.95 | 0.323 | 0.0014 |
| average correlation | 0.1514 | -0.0504 |
| first component correlation | 0.1579 | -0.0688 |
| first 5 components correlation | 0.1514 | -0.0504 |

For CD-SVD, the total KL-divergence between source and target domain ratings, the ratio between source and target domain densities, and the ratio of number of users to number of target domain items have significantly negative correlations with improvement ratio. The only significantly positive correlation is for the variance of source domain ratings.

Again, none of the CCA-related features have a significant correlation with the amount of improvement we get in the cross-domain recommender systems, compared to the single-domain one. The scatter plot of these features against the improvement ratio of algorithms can be found in Appendix C.4.

### 7.2.4 Summary

In this section, we studied the correlation of various data characteristics with RMSE, MAE, RMSE-based improvement ratio, and MAE-based improvement ratio of algorithms. In summary, we have seen that some of the data features are consistently correlated with the results in different datasets and some other are inconsistent.

For example, the total KL-divergence between the source and target domains, if significant, mostly has a negative correlation with the improvement ratio of algorithms. This means that the distribution of ratings in the source and target domains should be similar to each other to have a better result in cross-domain recommendations.

Also, the CCA-related features, if significant, have a positive correlation with the improvement ratio of algorithms. It means that stronger canonical correlation between the source and target domains is associated with better cross-domain recommendations. The only exception is the number of correlations $> 0.95$ in the Yelp dataset that can be ignored because of very few domain pairs that actually have a canonical correlation of $> 0.95$.

Some of the features had different behaviors in different datasets. For example, the correlation of number of users and average ratings of target users with MAE of CD-CCA is negative in the Yelp dataset, but positive in the Supermarket dataset; skewness of source domain ratings has a positive correlation with MAE of CD-SVD in Yelp dataset, but a negative one in the Supermarket dataset; Source density and variance in user-based KL-divergence between the source and target domains has a positive correlation with IR in the Supermarket dataset and a negative one in the Yelp dataset; and Kurtosis of target domain ratings is negatively correlated with RMGM's IR in the Yelp dataset and positively correlated with it in the Supermarket dataset.

For the Imhonet dataset, we see very different results. None of the CCA-related features have a significant correlation with the error of algorithms. Also, there are fewer significant correlations between the errors and the data features. Additionally, there is no shared factor that significantly correlates with the error of any two algorithms. We attribute these results to having unreliable CCA correlations for the Imhonet dataset. As we have mentioned in Section 4.2, the number of users is always smaller than the number of source or target domain items in the Imhonet dataset. This, results in less significance in the canonical correlations. As we have seen in this chapter and in Table 51 in Appendix C.1, the canonical correlations between domain pairs are high, but insignificant. Consequently, although we achieve better recommendations using CD-CCA, we cannot discover any significant correlations between the error of algorithms and the CCA-related features in Imhonet.

Another reason for such a different result in the Imhonet dataset is the few number of domain pairs. Compared to the 50 studied domain pairs in the Supermarket dataset, and the 158 domain pairs of Yelp dataset, the 12 domain pairs of Imhonet dataset are too few. The few number of domain pairs results in a few number of datapoints for the correlation analysis and leads to unreliable results for that.

Finally, we have only studied the Pearson correlation between the dataset characteristics and performance of algorithms. In case there is a non-monotonic relationship between the dependent and independent variables, Pearson correlation cannot capture that. As we have seen in the scatter plots presented in Appendix A.5, B.4, and C.4, some of the variables do not follow a monotonic association with the performance of algorithms. For example, in Figure 73, the scatter plot for average rating of target domain and error of SD-SVD, has an upside down $U$ shape.

## 7.3 REGRESSION ANALYSIS

Although correlation analysis can show us the general trend of relationship between two variables, it cannot determine how much each independent variable has a role in prediction of the dependent variable compared to other factors. To understand the relative importance of each of the cross and single-domain data features, explained in Section 7.1, we use multi-variable regression analysis. In the following sections, we explain our setup for the regression analysis and report its results on each of the datasets.

### 7.3.1 Experiment Setup

We run multi-variable regression on the dataset features, presented in Section 7.1, as independent variables. For the dependent variable, we choose RMSE and RMSE-based improvement ratio of each of the algorithms on each of the datasets.

Because of having so many features (33), we start with the features that show significant correlations with the algorithms in Section 7.2 for each dataset[1].

Since many of the aforementioned data features can be correlated to each other, we perform a multicollinearity analysis [5] and select a subset of presented features, as the independent variables. To do this, we sort the variables based on their correlations that are reported in Section 7.2. We normalize these variables and look at their condition index and variance inflation factors (VIFs). We remove the feature that causes the maximum condition index and/or has the maximum VIF and repeat the process for the rest of the variables. We continue this process until we reach an acceptable condition index (less than 30). Consequently, we end up with a different set of variables for each of the datasets.

We run the regression analysis on these sets of variables once with the RMSE of each algorithm as the dependent variable, and once with each algorithm's improvement ratio over SD-SVD as the dependent variable, in each of the datasets. The results are reported in the following sections.

### 7.3.2 Regression Analysis for the Supermarket Dataset

In this dataset, we end up with 10 variables after performing the multicollinearity analysis. These variables and the results of regression on the RMSE of algorithms are listed in Table 28. This table shows the coefficients of each of the variables, with stars representing their significance, the RMSE, $R^2$, and p-value for $R^2$ of the model.

[1] We tried starting the multicollinearity analysis with all of the variables. The results were similar to, or worse than starting with the significant set of variables.

Table 28: RMSE regression analysis results for the Supermaket dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001

| | CD-CCA | CD-SVD | RMGM | CMF | SD-SVD |
|---|---|---|---|---|---|
| intercept | 0.4244*** | 0.3917*** | 0.4287*** | 0.1095 | 0.4319*** |
| target mode rating | 0.0018 | 0.0412 | 0.0811* | -0.2233 | 0.0256 |
| mean user KL-divergence | -0.0129 | 0.0249 | -0.0349 | 0.1211 | -0.0143 |
| user to source item ratio | -0.1115*** | -0.0158 | -0.0111 | 0.2977 | -0.0099 |
| CCA correlation $\geq 0.95$ | -0.1316*** | -0.0268 | -0.0419 | 0.1632 | -0.0266 |
| target density | 0.0028 | -0.0566 | -0.2134*** | -0.138 | -0.0583 |
| target item size | 0.0149 | -0.0168 | -0.0661 | 0.005 | -0.025 |
| target skewness rating | -0.0135 | -0.0543 | 0.0794* | 0.4595** | -0.0567 |
| source to target density ratio | 0.0154 | 0.2252*** | 0.1584*** | -0.2605 | -0.0787 |
| user size | 0.0401** | 0.1261*** | -0.1154*** | 0.1354 | 0.1512*** |
| variance user KL-divergence | -0.0156 | -0.0439 | 0.1452** | 0.5374* | -0.1199* |
| RMSE | 0.0202 | 0.0514 | 0.0441 | 0.23 | 0.0474 |
| P value | 1.55E-06 | 1.12E-05 | 1.73E-19 | 1.66E-03 | 1.79E-08 |
| R2 | 0.572 | 0.52 | 0.913 | 0.351 | 0.666 |

Based on the reported $R^2$s, we can see that all p-values are significant. Also, we can see that although all p-values for the $R^2$s are significant, the p-values of all variables are not. For CD-CCA, we see negative significant relationships between the RMSE with the number of components with CCA correlation more that 0.95 and the ratio between number of users and number of source domain items. This means that as the CCA correlation increases, the error of CD-CCA decreases. Also, as we have a taller source domain rating matrix, we have less error in CD-CCA. However, an increase in the number of users by itself increases the error. For CD-SVD, the density ratio and number of users both increase the RMSE. This means that having a denser source domain, compared to target domain, we get more error in CD-SVD.

For SD-SVD, we see a positive relationship for the number of users and a negative one for the variance in user-based KL-divergences between the source and target domains. The latter relationship is meaningless since the source domain information is not used in SD-SVD algorithm.

In RMGM, we can see that the more skewed the target domain is, we will have more error. Also, the user-based KL-divergence variance, ratio between source and target domain densities, and mode of ratings in the target domain all have a positive relationship with the error. However, the denser the target domain is and the more the number of users is, the less error we have in RMGM.

The user-based KL-divergence variance and the target domain skewness have a positive relationship with CMF's error also.

The maximum significant coefficient variable belongs to the number of canonical correlations $>= 0.95$ for CD-CCA, source to target domain density rations for CD-SVD, number of users for SD-SVD, target domain density for RMGM, and the variance in user-based KL-divergence of domains for CMF.

In general, we can see that the variance of user-based KL-divergence and the target domain skewness are both positively related to CMF and RMGM errors; the number of users can have a positive or negative relationship with the RMSE of algorithms; and the density ratio between source and target domains have a positive relationship with the error of both RMGM and CD-SVD.

The relationships get more clear if we look at the improvement ratios in Table 29. Here, we see that variance in user-based KL-divergence is associated with less improvement in all of the cross-domain recommenders, compared to SD-SVD. This means that as the KL-divergences of each users' ratings between source and target domains varies more, using the source domain information helps less in cross-domain recommendations. The next important factor is the density ratio between source and target domains, which is significant for IR of RMGM, CD-SVD, and CD-CCA. The denser the source domain is, compared to the target domain, the less improvement we will have in the RMSE of these algorithms compared to SD-SVD. Skewness of ratings in the target domain has a negative effect on the IR of RMGM and CMF, in accordance with its relationship with the error of these algorithms. The number of users have a contradictory effect in CD-CCA, but its relationship with RMGM is consistent. Although it has a positive relationship with the RMSE of CD-CCA, it has a positive relationship with its IR too. In other words, although the more users we have, the more the error of CD-CCA will be, we will also see more improvement over SD-SVD with

Table 29: Improvement ratio regression analysis results for the Supermaket dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| | CD-CCA | CD-SVD | RMGM | CMF |
|---|---|---|---|---|
| intercept | 0.0685 | 0.1671 | 0.06 | 1.2936 |
| target mode rating | 0.0063 | -0.0795 | -0.3454* | 0.4984 |
| mean user KL-divergence | -0.0118 | -0.161* | 0.0601 | -0.4848 |
| user to source item ratio | 0.2434* | 0.0464 | -0.0177 | -0.9167 |
| CCA correlation $\geq 0.95$ | 0.2415 | 0.0163 | 0.0094 | -0.5688 |
| target density | -0.1324 | -0.0589 | 0.4745** | 0.0809 |
| target item size | -0.1064 | -0.0295 | 0.1086 | -0.1517 |
| target skewness rating | -0.1239 | -0.0147 | -0.4594** | -1.9305** |
| source to target density ratio | -0.2716** | -1.0848*** | -0.7398*** | 0.6203 |
| user size | 0.2434*** | 0.1104 | 0.6934*** | 0.0982 |
| variance user KL-divergence | -0.3685** | -0.3047** | -0.9634*** | -2.5057** |
| RMSE | 0.1023 | 0.0972 | 0.1911 | 0.7658 |
| P value | 2.62E-12 | 4.74E-18 | 4.59E-19 | 1.84E-05 |
| R2 | 0.793 | 0.897 | 0.909 | 0.506 |

Table 30: RMSE regression analysis results for the Yelp dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| | CD-CCA | CD-SVD | RMGM | CMF | SD-SVD |
|---|---|---|---|---|---|
| intercept | 0.4889*** | 0.5221*** | 1.9023*** | 0.2964 | 0.5979*** |
| source kurtosis rating | -0.0022 | 0.1295 | -0.0134 | 0.1188 | 0.0287 |
| first component correlation | -0.0099 | -0.105* | -0.4184*** | -0.4066 | -0.0746 |
| target mode rating | 0.2317*** | 0.2401*** | 0.2979*** | 0.2769 | 0.2486*** |
| target density | -0.2123** | -0.3217*** | 0.1057 | -0.3196 | -0.3424*** |
| target median rating | 0.1951*** | 0.1286** | 0.078 | 0.2482 | 0.1379** |
| target skewness rating | 0.8091*** | 0.9822*** | -0.4937** | 1.7259 | 0.9036*** |
| RMSE | 0.1213 | 0.1317 | 0.1673 | 0.9004 | 0.1381 |
| P value | 2.33E-18 | 2.81E-19 | 1.06E-35 | 2.86E-01 | 6.50E-18 |
| R2 | 0.449 | 0.459 | 0.682 | 0.00949 | 0.435 |

larger number of users. This is because the error of SD-SVD is also positively correlated with the number of users.

In addition to these relationships, IR of CD-CCA improves as we have a taller source domain rating matrix, and IR of CD-SVD improves as the average user-based KL-divergence of the two domains decreases, and thus there is more similarity between average user rating distributions. Also, as the mode of target domain ratings increases, which can be an indicator of skewness of ratings, the IR of RMGM decreases.

### 7.3.3 Regression Analysis for the Yelp Dataset

Table 30 shows the results of regression analysis on the Yelp dataset. As we can see here, the RMSE of the models are more than in the Supermarket dataset and the p-value of $R^2$ for CMF is insignificant.

Also, none of the factors are significant in the CMF model. However, skewness of ratings in the target domain and mode of ratings in the target domain both have significant relationships with the error of RMGM, CD-CCA, CD-SVD, and SD-SVD. Mode of ratings in the target domain has a positive relationship with all of the RMSEs and skewness of

Table 31: Improvement ratio regression analysis results for the Yelp dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| | CD-CCA | CD-SVD | RMGM | CMF |
|---|---|---|---|---|
| intercept | 0.1249 | 0.0654 | -1.1226*** | 0.1734 |
| source kurtosis rating | 0.0126 | -0.0633* | -0.015 | -0.0409 |
| first component correlation | 0.0355 | 0.0242 | 0.3663*** | 0.3551 |
| target mode rating | 0.0207 | 0.0075 | 0.0077 | 0.0296 |
| target density | -0.2836*** | -0.0669** | -0.7502*** | -0.1731 |
| target median rating | -0.0762 | 0.0064 | 0.0171 | -0.1305 |
| target skewness rating | -0.0521 | -0.0762 | 1.1694*** | -0.6705 |
| RMSE | 0.1266 | 0.05 | 0.2111 | 0.6657 |
| P value | 1.10E-03 | 4.74E-04 | 6.66E-24 | 6.79E-01 |
| R2 | 0.103 | 0.113 | 0.539 | -0.0133 |

ratings in the target domain has a positive relationship with all RMSEs, but RMGM's. Target domain's density has a negative relationship with the errors of CD-CCA, CD-SVD, and SD-SVD and the median of target domain ratings has a positive relationship with them.

Looking at the regression results for improvement ratios in Table 31, we see no significant coefficients for CMF. The density of target domain has a negative effect on the improvement of RMGM, CD-CCA, and CD-SVD, over SD-SVD; the kurtosis of ratings in the source domain has a negative relationship with CD-SVD's IR; the skewness of ratings in the target domains has a positive relationship with RMGM's IR; and the canonical correlation of the first discovered component has a positive effect on RMGM's IR.

### 7.3.4 Regression Analysis for the Imhonet Dataset

Table 32 shows the results of regression analysis in Imhonet dataset. We can see that although RMSEs of models are low, the p-value of $R^2$ in CD-CCA is insignificant and none of the coefficients are significant in this model.

For SD-SVD and CD-SVD, we see a positive relationship between their RMSEs and kurtosis of ratings in the source domain, the ratio of users to source domain items, and

Table 32: RMSE regression analysis results for the Imhonet dataset; *: significant with p-value $< 0.05$; **: significant with p-value $< 0.01$; ***: significant with p-value $< 0.001$

| | CD-CCA | CD-SVD | SD-SVD |
|---|---|---|---|
| intercept | 0.2182*** | 0.1618** | 0.1794** |
| source to target density ratio | -0.0091 | 0.006 | -0.0224 |
| target var rating | 0.0071 | -0.0141 | 0.0626 |
| source item size | 0.0134 | -0.0783** | -0.0433 |
| source kurtosis rating | 0.0096 | 0.1116** | 0.0815* |
| user to target item ratio | 0.0049 | 0.3044*** | 0.1286** |
| user to source item ratio | -0.0073 | 0.2614*** | 0.2398** |
| RMSE | 0.0129 | 0.0121 | 0.017 |
| P value | 1.83E-01 | 3.22E-04 | 7.31E-03 |
| R2 | 0.425 | 0.961 | 0.86 |

the ratio of users to the target domain items. For SD-SVD, the first two relationships are meaningless because it does not use source domain information. Also, as the number of items in the source domain increases, the error of CD-SVD decreases.

For regression on the improvement ratios, we report the results in Table 33. Here, non of the p-values of $R^2$s are significant. But, we see a significant positive relationship between the ratio of users to source domain items and CD-CCA's IR; and a negative relationship between the ratio of users to the target domain items and CD-SVD's IR. This means that the taller the source domain rating matrix is, the more improvement we have in CD-CCA over SD-SVD; and the fatter the rating matrix in the target domain is, the less improvement we have in CD-SVD over SD-SVD.

### 7.3.5 Summary

As we can see in the regression results, there are many different factors that are important in each of the algorithms and each of the datasets. For example, the number of users is an important factor only in the Supermarket dataset, or the first component's CCA correlation is only important in the Yelp dataset.

Table 33: Improvement ratio regression analysis results for the Imhonet dataset; *: significant with p-value < 0.05; **: significant with p-value < 0.01; ***: significant with p-value < 0.001

| | CD-CCA | CD-SVD |
|---|---|---|
| intercept | 0.0774 | 0.0252 |
| source to target density ratio | -0.003 | -0.0677 |
| target var rating | 0.0883 | 0.2254 |
| source item size | -0.1218 | 0.0904 |
| source kurtosis rating | 0.1118 | -0.0714 |
| user to target item ratio | 0.2152 | -0.4745* |
| user to source item ratio | 0.4171* | -0.0414 |
| RMSE | 0.0582 | 0.0774 |
| P value | 6.84E-02 | 1.17E-01 |
| R2 | 0.636 | 0.535 |

There are some common important factors between the datasets also. For example, the density ratio of source and target domains appears in both Supermarket and Imhonet regression models and the source domain kurtosis appears in both Yelp and Imhonet dataset models.

However, there are two contradictory results between the datasets. The first one is related to the density of target domain. It has a positive relationship with RMGM's improvement ratio in the Supermarket dataset and a negative relationship with it in the Yelp dataset. It means that in the Supermarket dataset, the denser the target domain is, the more improvement RMGM has over SD-SVD. But, in the Yelp dataset, the denser target domain contributes to less improvement of RMGM's results.

The second one is the skewness of ratings in the target domain. In the Supermarket dataset, it has a positive relationship with RMGM's error and a negative relationship with its improvement ratio. However, this relationship works in the reverse direction in the Yelp dataset. This can be one of the reasons that leads to the good performance of RMGM in the Yelp dataset compared to the Supermarket dataset. Also, we can see that the mode of ratings in training data has a positive relationship with RMGM's RMSE. We know that in highly skewed datasets, the mode of ratings moves in the direction of skewness. So, if the ratings are skewed towards higher ratings (which is the case for most explicit feedback recommender systems' datasets), the mode is also going to be higher. Although there is no direct correlation and collinearity between the mode and skewness of Yelp, we think that this general rule can explain some of the contradiction that we see in the regression analysis of the Yelp dataset. Basically, we hypothesize that some of the "positivity" of the relationship between skewness and RMSE of RMGM in the Yelp dataset, is absorbed by the positive relationship between the mode of target ratings and RMGM's RMSE. The same can be true for the median of target domain ratings, that appears in the Yelp dataset's regression analysis.

As for the CCA-related features, we see the number of correlations that are $> 0.95$ in the Supermarket dataset's regression analysis and the canonical correlation of first component in the Yelp regression analysis. We can see that the direction of their relationship, when significant, is as expected: to lower the cross-domain recommenders' error and to increase their improvement ratios. However, they are not present in the Imhonet dataset's regression analysis.

Also, we should note the number of data points in the analysis. Each domain pair is one datapoint in this regression analysis. So, we have 12 data points for the Imhonet dataset, 50 in the Supermarket dataset, and 158 in the Yelp dataset. These number of datapoints, especially for Imhonet, are not nearly enough for having a powerful regression analysis.

Eventually, we only looked at the possible linear relationships among the dependent and independent variables. Thus, we cannot find other kinds of possible relationships, such as polynomial or exponential ones. Looking at the scatter plots of independent variables and error of algorithms in appendices, we can see that most of the independent variables do not have a strict linear relationship with the dependent variables.

## 7.4 NATURE OF GOOD DOMAIN-PAIRS

In this part of our analysis, we look at the specific samples of domain pairs to get a deeper understanding of the results of previous analyses and answer our research question Q2.3.

More specifically, we look at the domain pairs with a high CCA to see if they can obtain a higher improvement in cross-domain recommenders versus the single-domain recommender. We examine the domain-pairs with a high CCA and low improvement in a closer look to understand the reason behind this behavior. As a reverse look at these results, we look at the domain pairs with a high improvement ratio and their characteristics. More specifically, we look at the domain-pairs with a high improvement in cross-domain recommenders versus the single-domain recommender, and a low CCA, to understand the other factors that affect this result.

First, we look at the domain pairs that have a high canonical correlation between the domain pairs and are having good results in cross-domain recommenders. In the Supermarket dataset, we see very good results in cross-domain recommendation versus single-domain recommendation in domains such as “international food → fruit and vegetables” and “dairy → bread”. In these domain pairs, all of the cross-domain recommender systems perform significantly better than SD-SVD. Also the canonical correlation of them (calculated by the average of CCA for the first five components) is in the first 10 and 25 percentile of all domain pairs in the dataset, respectively. In the Yelp dataset, examples include “active life → food”, “food → restaurants”, and “home services → local services”. In the same way, all of the cross domain recommenders perform significantly better than SD-SVD in these domain pairs and their canonical correlation is in the first 5, 1, and 33 percentile of all canonical correlations in the Yelp dataset.

Next, we look at the domain pairs with low CCA results that have high errors in cross-domain recommendations. In the supermarket dataset, examples of these domain pairs are “bread → events”, “canned and pickled → home outdoor”, and “fruit and vegetables → events”. In these domain pairs, the single-domain algorithm has significantly less MAE and RMSE compared to all of the cross-domain algorithms. Accordingly, the canonical correlation between the domains are in the last 6, 30, and 25 percentile of all canonical correlations of this dataset. To illustrate such relationship in the Yelp dataset, we can name “active life → hotels and travel” and “arts and entertainment → pets” domain pairs. The canonical correlation between these domain pairs are in the last 40 percentiles of the Yelp dataset’s canonical correlations. Similarly, SD-SVD performs better than all of the cross-domain algorithms in these domain pairs.

After looking at the domain pairs with expected results in CCA and performance of algorithms, we examine the ones that act in a reverse order. Thus, we look for the domain pairs with a relatively low canonical correlation between the domains, but good cross-domain results. For example, in the Supermarket dataset, “events → fruit and vegetables” has a very good cross-domain results: all of the cross-domain algorithms perform significantly better than SD-SVD in this domain pair. However, this domain pair’s CCA in in the last 40 percentile of all CCA’s in the dataset. Looking at other characteristics of this domain pair, we can see that the density ratio of source to target domain is the minimum for this domain pair in the dataset. This is associated with a high improvement ratio and thus better performance of cross-domain recommenders. Also, the number of users and items in the target domain are very close to each other in this domain pair. This, can result in unreliable CCA results. Looking at the number of components with a significant CCA in this domain pair, we can see that less than 41% of the component correlations are significant. As examples of this phenomenon in the Yelp dataset, we can name “mass media → local flavor” and “professional services → religious organizations”. In both of these domain pairs, the cross-domain algorithms all perform significantly better than SD-SVD, but the canonical correlations are in the last 4 percentile in the dataset. In “mass media → local flavor”, the density ratio of source to target domain is high which is associated with less RMSE in the Yelp dataset. Also, we have less skewness in the “local flavor” domain (least 8 percentile) that is associated with a better result in cross-domain recommendations. Again, the number of users and target items is very close to each other and makes the CCA results unreliable. Less than half of the discovered components have a significant canonical correlation. In “professional services → religious organizations” the target domain’s density is very high (high 2 percentile) that is associated with less error for the cross-domain algorithms. Also, there are only 9 users, 9 source items, and 8 target items in this domain pair that causes less reliability in CCA results. Only one of the six canonical components have a significant correlation in this domain pair. In Imhonet, “book → movie” has a relatively low CCA. But both cross-domain algorithms work significantly better than SD-SVD in this domain pair. Generally, we believe that we cannot trust the CCA results in this dataset because of the very high sparsity of the data. As we have seen, none of the correlations for discovered components in any of the domain pairs are significant.

In some of the domain pairs of some of the datasets, we can see a high canonical correlation, but high error of cross-domain recommenders, compared to the single-domain one. For example in the Yelp dataset, "food → hotels and travel" has a high CCA (high 20 percentile), but SD-SVD works better than two of the four cross-domain algorithms in this pair. One reason can be the high skewness of ratings in the "hotels and travel" domain compared to other domains (high 20 percentile). Also the density of target domain is in the lower side (low 30 percentile) and the variance of user-based KL-divergence in the source and domain pairs are high that can result in more RMSE and less improvement of cross-domain recommenders. In the imhonet dataset, the canonical correlation in "perfume → game" is high, but CD-SVD is not significantly better than SD-SVD for this domain. Also, the variance of error is very high in this domain pair. Again, for the Imhonet dataset, we cannot rely on CCA results, or correlation and regression analyses, because of the sparsity of this dataset and the few number of domain pairs in it.

Another interesting observation is that the relationship between the domain pairs is not always reciprocal. For example, in the Supermarket dataset, the "gifts" domain helps in improving the results for the "bread" domain, but adding the "bread" domain information to the "gift" domain does not help. Also, the "home outdoor" domain helps the "fruit and vegetable" domain, but the reverse is not true. In the Yelp dataset, the "nightlife" domain information improves the recommendations in the "restaurants" domain. But, the "restaurants" domain information does not add much to the "nightlife" recommendation results. On the other hand, the "health and medical" domain information helps the recommendations in the "nightlife" domain, but, the "nightlife" domain does not help the recommendations in the "health and medical" domain.

Finally, there are some domain pairs that work surprisingly well together. For example, one may not intuitively think that there is any semantic relationships between the "international food" and "home cleaning" domains in the Supermarket dataset. However, the canonical correlation of "international food → home cleaning" is relatively high and the cross-domain algorithms have significantly less errors compared to SD-SVD in this domain pair. The same happens for the "automotive → nightlife" and "health and medical → arts and entertainment" domain pairs. One may not see any relationship between user tastes in arts and their reviews on hospitals, or user reviews on night clubs and where they repair their car. But, we can see that both the canonical correlation and improvement ratios of cross-domain algorithms are high in these two domain pairs. This means that CCA can capture some unexpected, however existing, relationships between the domains.

On the other hands, there are some domain pairs that intuitively seem to be similar, but cross-domain recommenders do not work better than single-domain ones in them. For example, one may think that user preferences in the "active life" domain is related to their preference in the "hotels and travel" domain. However, the "active life → hotels and travel" domain pair has a low canonical correlation and adding the user ratings of "active life" domain to the "hotels and travel" domain worsens the recommendations on "hotels and travel". Thus, we cannot only rely only on the "intuitive similarity" of domains in choosing them for cross-domain recommendations.

## 7.5 SUMMARY

In this chapter we researched on the data characteristics that can lead us towards selection of better domain pairs for cross-domain recommendations. Our goal was to answer research question Q.2, to find the factors that distinguish between a beneficial domain pair and a non-helpful one, and determine the amount of improvement in cross-domain recommenders. Also, we aimed to find out more about the nature of good domain pairs and study if they match our expectations based on intuitively similar domain pairs.

We first defined the data features in Section 7.1, based on dataset general statistics, central tendency and dispersion descriptive statistics of ratings in each of the domain pairs, divergence of rating distributions in domain pairs, and CCA-related features.

Then, we performed a correlation analysis to understand the general trend between the performance of algorithms and these data features. For measures of performance, we studied the RMSE and MAE of algorithms. For measures of improvement, we analyzed the ratio of improvement of cross-domain algorithms compared to the single-domain algorithm, calculated based on RMSE and MAE (IRs). We found out that the correlation values and their significance varies among different datasets. For example, in the Supermarket dataset, the denser the target domain is, the more improvement we will have in cross-domain recommendations. But, this correlation works in reverse in the Yelp dataset. Even in the same dataset, some of the data characteristics had a different effect on different algorithms' results. For example, in the Yelp dataset, the improvement ratio of RMGM algorithm decreases by more skewness in the target domain's ratings. However, more skewness is associated with more improvement ratio in the CMF algorithm.

Among all of the data characteristics, the CCA-related features and the divergence-related features performed most consistently among the datasets. Mostly, the CCA-related features had a positive correlation with improvement ratio of algorithms and a negative one with the error of algorithms. Also, the KL-divergence between all ratings in the source and target domains mostly have a negative correlation with the improvement ratio of cross-domain algorithms.

To uncover the relative linear relationship of these data characteristics with error and improvement ratios of algorithms, we performed regression analysis in Section 7.3. Before that, we looked at the multicollinearities among the variables and removed the problematic features. We ended up with different variables in each of the datasets. Then, we modeled the RMSE and RMSE-based improvement ratio of algorithms with these features in a linear model. In summary, the regression model usually did not explain all of the variability in the data; which is expected. Because each of the algorithms have a complicated approach to estimate user ratings and this complicated task cannot be completely modeled with a linear regression analysis. Based on the regression results, we encountered with some contradictory relationships between the features and our dependent variables. On the other hand, we discovered that CCA-related features, in both Yelp and Supermarket datasets, work as we expect: if they have a significant relationship, they work towards less error and more improvement ratio. Although the size of the effect is very small in these factors.

Finally, we studied the domain pairs that had unexpected behaviors: the ones with high CCA and low cross-domain performance, the ones with low CCA and good cross-domain performance, the ones which are intuitively close to each other but had a bad cross-domain performance, and the ones that had a good cross-domain performance but were not intuitively close to each other. In the cases with low CCA and high performance, or vice versa, we have seen that other features of domain pairs are working towards getting these surprising results. Also, in most of them, CCA results were not as reliable as we would have expected.

As a whole, we conclude that different variables are effective for each of the algorithms in each of the datasets and we achieve different results in each setting. Some of the differences are because of the different nature of each of the datasets. For example, the Supermarket dataset is denser than the other two datasets and is less skewed. It does not reveal the "rating" of users on the items, but their purchase history. On the other hand, the Imhonet dataset is very sparse. As a result, the canonical correlations in this dataset are not significant. However, the canonical correlation between all domain pairs are very high in this dataset. To illustrate, the least value for average of canonical correlation in the first five components in Imhonet is more than 0.8; while for Yelp, the minimum value for this factor is less than 0.5 and for the Supermarket dataset, it is less than 0.3.

Also, we conclude that CCA-related features, although not always significant, and although having a small effect size, work mostly similar in different datasets and algorithms. Additionally, based on the surprising domain pairs that we studied in Section 7.4, we decide that CCA can capture some relationships that may not be intuitively meaningful to us, but lead to better cross-domain recommendations. Yet, CCA alone may not be sufficient in finding the best matches for some domains. Other, dataset-dependent variables, can change the way two domains work together in cross-domain recommendations.

# 8.0 AUXILIARY DOMAIN CLASSIFIER

This chapter of dissertation aims to find answers to research question Q3, defined in Section 1.2. More specifically, we would like to examine if we can classify the domain pairs based on the data characteristics defined in Section 7.1. The dependent variable for this classification is if a domain pair is an appropriate pair for cross-domain recommendation or not. In other words, if it is beneficial, in terms of the error of recommendation results, to use cross-domain recommenders in each domain pair; or having the single-domain recommendation on the target domain is good enough.

Having such classifier in hand, one can decide on the usefulness of an auxiliary domain, for a specific target domain, before performing the cross-domain and single-domain recommender algorithms. In the next sections, we first explain the setup under which we run the experiments, then we present the results of experiments, and summarize them.

## 8.1 EXPERIMENT SETUP

As we have concluded in the previous chapters, and we can see in the figures in Appendix A.5, B.4, and C.4, the more improvement ratio of cross-domain algorithms does not always mean that there is a significant improvement of cross-domain algorithm over the single-domain one. Thus, we do not select error or improvement ratio as our dependent variables in this chapter. Since we would like to discover the domain-pairs that have better results in cross-domain recommender systems, compared to the single-domain ones, the dependent variable in this classifier should represent such characteristic. The choice of dependent variable for such classifier is based on our goals to study the feasibility of detecting beneficial domain pairs. A domain pair is beneficial, using a specific cross-domain algorithm, if adding the information of the source domain to the target domain and applying such algorithm provides us with better results compared to using the single-domain recommender algorithm only on the target domain data. As a result, we choose a binary variable (SigIndx) that indicates if a cross-domain recommender has performed significantly better than the single-domain recommender. To be more clear, if $\ll$ represents the "significantly less" relation between error of algorithms[1], and $a_i$ represents one of the cross-domain algorithms, then Equation 8.1 explains the dependent binary variable calculated over SD-SVD algorithm. We use the RMSE of algorithms as the measure of error.

$$\text{SigIndx}_{a_i,\text{SD-SVD}} = \begin{cases} 1, & \text{if } \text{Error}_{a_i} \ll \text{Error}_{\text{SD-SVD}} \\ 0, & \text{otherwise} \end{cases} \tag{8.1}$$

Note that this is one possible definition of gold-standard for class labels and the classifier that we can use. It is possible to have other definitions for SigIndx. For example, the domain pairs that work significantly better than SD-SVD in all cross-domain algorithms. However, we do not experiment on this definition because of the diverse results that we are getting from different cross-domain algorithms. In other words, each of the cross-domain algorithms have their own strength and weaknesses that relate to domain-pair characteristics. Thus, the number of domain-pairs in which all of these algorithms work significantly better than SD-SVD is very limited.

Also, it is possible to have a multi-class classifier instead of a binary classifier. For example, we can train the classifier based on the set (or number) of algorithms in which a domain pair performs significantly better than SD-SVD. However, since the focus in this dissertation is to study the feasibility of distinguishing between beneficial and non-beneficial domain pairs, we focus on the simple case of a binary classifier for each of the algorithms.

Since we have four different cross-domain algorithms, we train the classifier four times, every time based on the performance of one of them versus SD-SVD performance.

To test the generalizability of the trained classifiers, we explore three general setups: *a*) having the test and train domain pairs from the same dataset to see how accurate the

[1] p-value $< 0.05$

classifier is within a specific domain nature or system; *b*) having the train domain pairs from one dataset and the test domain pairs from another dataset to see how much this classifier can be accurate on other domains with a different nature and how much does this meta-knowledge transfers between the domains with various nature; and *c*) mixing the domain pairs of two datasets with different natures and training a classifier on this mix to see the effect of a this merge on their results.

We use 5-fold cross-validation and repeat the experiments for five times. So, in general, we run the experiments 25 times on each dataset.

For the independent variables, we remove the central tendency features because they had less variability and added to the error of classifiers. We use all of the rest of features as dependent variables. However, to better understand the role of CCA-based features in classifying the good domain pairs, we run the experiments with three different sets of independent variables: *a*) using all of the data features; *b*) using all of the data features except the CCA-related features; and *c*) using only the CCA-related features. By this setup, we want to understand if the CCA-related features play an important role in the classification of good domain pairs or not. We compare these setups with a random classifier as baseline. To be more specific, we calculate the probability of misclasification, if we assign each domain pair to a class in random. We use the dataset priors for this random classification. For example, if the positive class happens in 70% of the times in the dataset, we calculate the probability of misclassification based on a random classifier that returns a positive label in 70% of the times.

Eventually, we use linear Support Vector Machine (SVM) classifier to run the experiments[2].

To evaluate the classifier, we use classifier error as our loss function. In other words, we count the number of misclassified domain pairs as the loss of SVM.

In the following sections, we present our results in each of the experiment settings for all of the datasets.

[2] We tried the RBF and polynomial kernels, but the results of linear classifier were better. Also we use the "fitsvm" function in Matlab as the implementation of SVM.

Table 34: Percentage of positive class labels in each dataset

| | Supermarket | Yelp | Imhonet |
|---|---|---|---|
| CD-CCA | 70 | 48.73 | 100 |
| CMF | 14 | 46.2 | N/A |
| RMGM | 72 | 12.03 | N/A |
| CD-SVD | 8 | 5.7 | 16.67 |

### 8.2 WITHIN DATASET CLASSIFICATION

In this section, we experiment on the classification of beneficial and non-beneficial domain pairs in each of the datasets. We run 5-fold cross validation over the 158 domain pairs in the Yelp dataset, 50 domain pairs in the Supermarket dataset, and 12 domain pairs in the Imhonet dataset.

Table 34 shows the percentage of these domain pairs that have a significantly less RMSE compared to SD-SVD, and thus, have a positive class label.

We run SVM on each of these datasets with three independent variable sets and a random baseline. The results of this classification on the Supermarket dataset is shown in Figure 43.

As we can see in this picture, the SVM classifier performs better than the random labeling in almost all of the algorithms with most of the variable sets. In predicting the classification labels for CD-SVD, the CCA-related set of features performs better than the random class label assignment. However, using all of the variables or all, except CCA-related ones perform similar to random assignment of class labels. Although relying on only CCA-related features have the best performance for CD-SVD, for the rest of the algorithms, adding or removing the CCA-related features in the set of independent variables, does not change the loss of classifier significantly.

Figure 44 shows the results of running SVM on the Yelp dataset. In this dataset, the SVM classifier always performs better than random class assignment. Both CCA-related

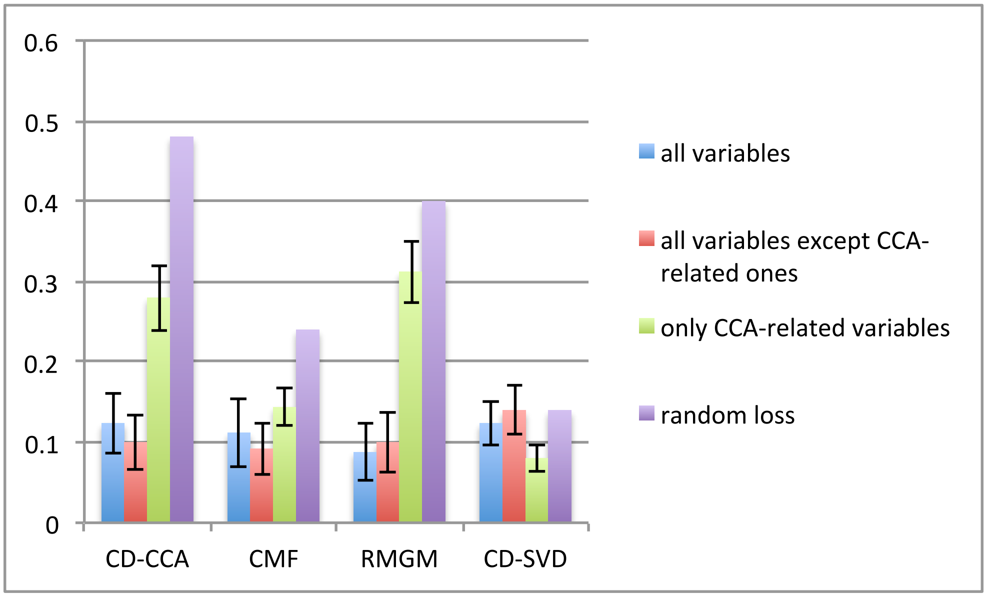


Figure 43: Classification loss in the Supermarket dataset

features and data features except the CCA-related ones, build the best classifier for CD-SVD. However, in classification of the other algorithms' results, although classifying the domain pairs based on only CCA-related features performs better than random assignment, adding CCA-related features does not significantly change the classification loss. So, in these cases, we can predict if a domain-pair will be beneficial in the cross-domain setting, by looking at the rest of the data features.

Figure 45 shows the classifier loss on the significance of error on domain pairs in CD-SVD algorithm in the Imhonet dataset. Since we always have a significantly better results in CD-CCA, compared to SD-SVD, in this dataset, all the losses for this algorithm is going to be zero. So, we do not show it here. In this picture, we can see that the SVM built on only CCA-related features and all features are doing better than the random classifier. However, having all of the features except for CCA-related ones does not perform significantly better than the random classifier.

In summary, we see that in all of the datasets, we can predict if a domain pair is suitable for cross-domain recommendation or not using the data features. Our prediction works

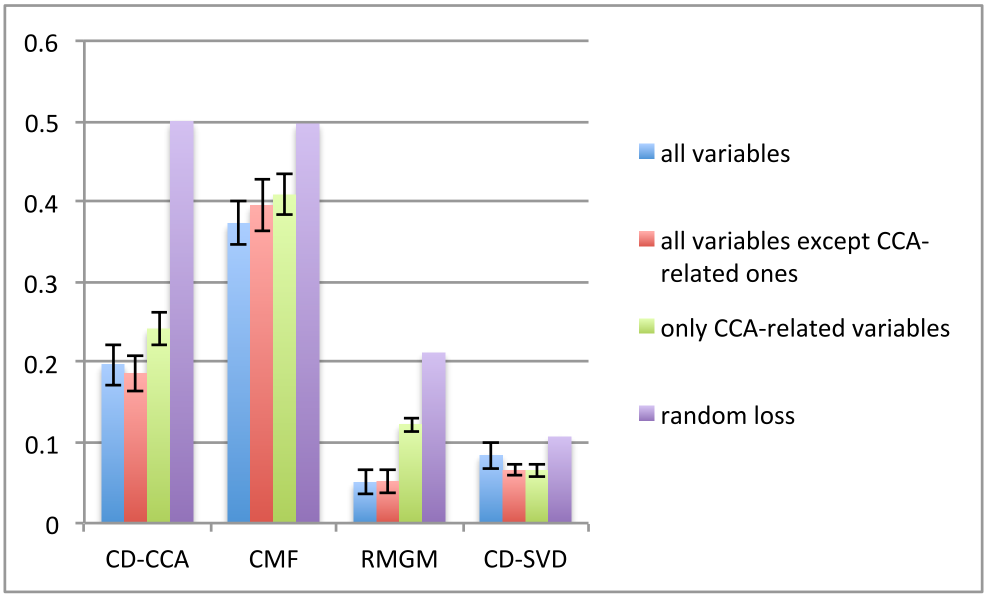


Figure 44: Classification loss in the Yelp dataset

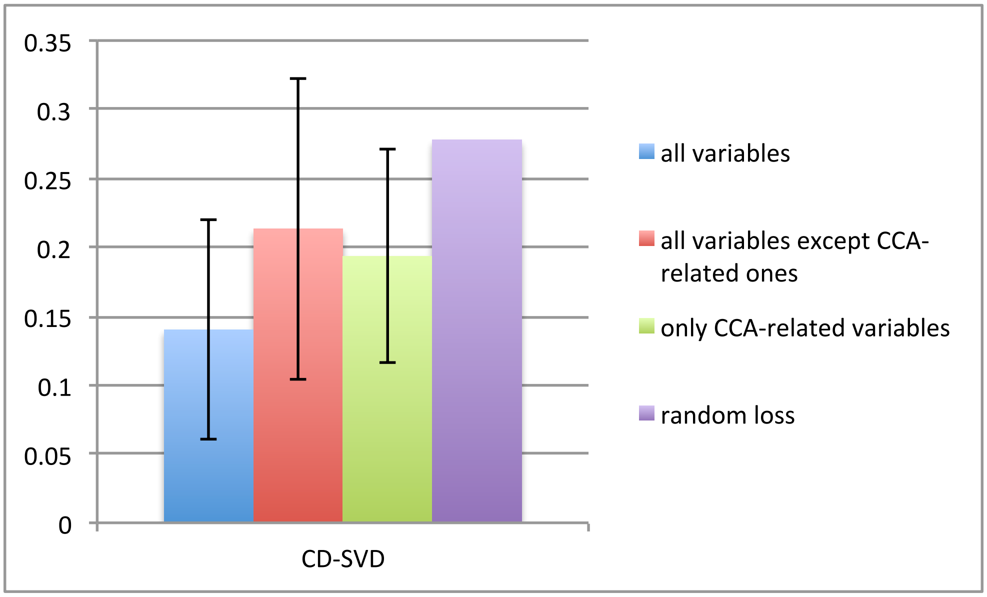


Figure 45: Classification loss in the Imhonet dataset

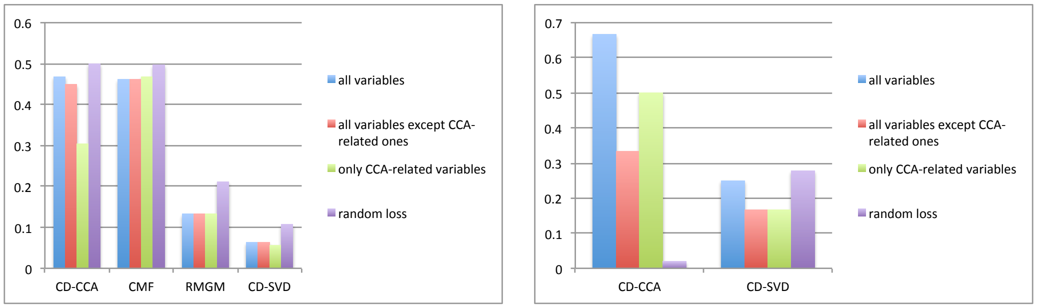
better than a random prediction that is based on the dataset priors. Also, we have seen that CCA-related features are important to classify domain pairs for the CD-SVD algorithm. However, for the rest of the algorithms, adding these features does not significantly change the results of a classifier that is built upon the rest of the features. But, it does not harm the results either.

## 8.3 CROSS DATASET CLASSIFICATION

In this section, we test the generalizability of the classifier for domain pairs, by training it on one dataset and testing it on another dataset. Again, we use three sets of independent variables and compare the classifier's results with a random classifier.

Figures 46a and 46b show the results of classifiers trained on the Supermarket dataset and tested on the Yelp and Imhonet datasets. As shown in these figures, except for CD-CCA in Figure 46b, the cross-dataset classifier performs better than the random classifier, that is based on the target dataset priors. The reason that SVM performs bad in comparison with the random classifier for CD-CCA in Imhonet is that there is no negative class labels for CD-CCA in that dataset. This happen because CD-CCA always performs significantly better than SD-SVD in the Imhonet dataset. In the rest of the experiments, we can see that the classifiers trained on CCA-related features only perform better, or similar to the classifiers trained on the rest of the dataset features, or all of the dataset features. Especially in the classifier for CD-CCA in Yelp and for CD-SVD in Imhonet, we can see that the classifier trained on CCA-related features of the supermarket dataset performs much better than the other classifiers.

Figures 47a and 47b show the loss of the classifiers trained on the results of algorithms in the Yelp dataset and tested on the Supermarket and Imhonet datasets. Again, we can see that the cross-dataset classifier that is trained on CCA-related features only performs the best and better than the random classifier in all of the settings (The CD-CCA data in the Imhonet dataset does not have a negative class label, and thus is not reliable). For the classifiers trained on all of the variables, we mostly see better results, compared to the



(a) Classification loss for each of the algorithms with the classifier trained on the Supermarket dataset and tested on the Yelp dataset

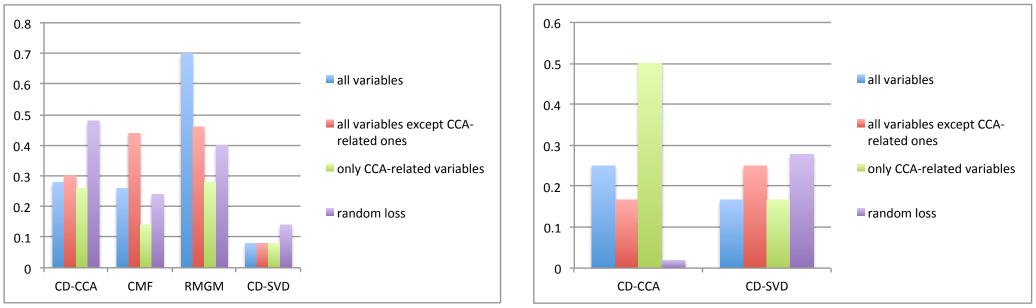
(b) Classification loss for each of the algorithms with the classifier trained on the Supermarket dataset and tested on the Imhonet dataset

Figure 46: Classification loss for cross-dataset experiments; trained on the Supermarket data

random baseline. Also this classifier is mostly better than the one trained on all, except CCA-related, features. Only in RMGM results tested on the Supermarket dataset, the all-variables classifier performs very poor, and even worse than the random baseline.

In figures 48a and 48b we see the loss of the classifiers trained on the results of algorithms in the Imhonet dataset and tested on the Supermarket and Yelp datasets. As for the previous experiments, we cannot trust the CD-CCA results, because the classifier trained on the results of this algorithm in the Imhonet dataset always returns the positive label. For CD-SVD, we can see that the SVM trained on CCA-related features performs better than all other classifiers. The other two classifiers even perform worse than the random baseline.

Eventually, these experiments show that the classifiers trained on the domain pairs in one dataset can mostly be transferable to the domain pairs of other datasets. However, the classifiers trained only on the CCA-related features are the ones that perform the best in this transfer.



(a) Classification loss for each of the algorithms with the classifier trained on the Yelp dataset and tested on the Supermarket dataset

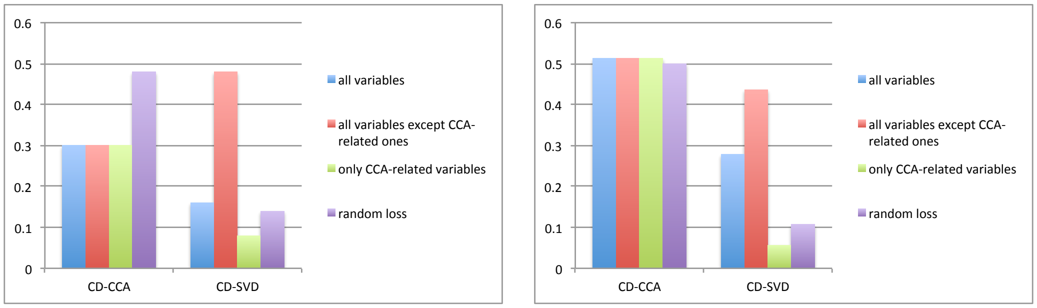
(b) Classification loss for each of the algorithms with the classifier trained on the Yelp dataset and tested on the Imhonet dataset

Figure 47: Classification loss for cross-dataset experiments; trained on Yelp data

## 8.4 MIXED DATASET CLASSIFICATION

In this section, we experiment on transferability of domain pair classifiers by training them on a set of domain pairs coming from mixed datasets. To be more specific, we select each two datasets, merge the domain pairs features and the results of algorithms on them together, and run the classification experiments with 5 fold cross-validation on this mixed dataset. Consequently, the test and train domain pairs come from both of these datasets. As a result of this mixing, the percentage of positive labels in the new merged datasets are going to be different with the original ones, and thus, the performance of random baseline will change with that. Table 35 shows the percentage of positive class labels for each of the algorithms in each of the datasets.

Figure 50 shows the loss of SVM classifiers for the mix of Imhonet and Supermarket datasets. As we can see here, the classifiers trained on CCA-related features are all performing significantly better than the random baseline and better than the other classifiers. In most of the cases, the classifiers trained on all data features, and the ones trained on all, but CCA-related, features are also performing better than the random baseline. The only



(a) Classification loss for each of the algorithms with the classifier trained on the Imhonet dataset and tested on the Supermarket dataset

(b) Classification loss for each of the algorithms with the classifier trained on the Imhonet dataset and tested on the Yelp dataset

Figure 48: Classification loss for cross-dataset experiments; trained on Imhonet data

case in which a classifier performs worse than the random baseline is the SVM trained on all features, for the CD-CCA algorithm.

Figure 50 shows the loss of classifiers on the mix of Supermarket and Imhonet datasets. Here, since because of the merge of the datasets, the label for CD-CCA algorithm is not always positive, we can actually test the classifiers' performance for this algorithm. As shown in the figure, in the classifiers trained on CCA-related features perform significantly better than all other classifiers and the random baseline, in both CD-CCA and CD-SVD algorithms.

Table 35: Percentage of positive class labels in each mixed dataset

| | Supermarket + Yelp | Yelp + Imhonet | Supermarket + Imhonet |
|---|---|---|---|
| CD-CCA | 53.84 | 52.35 | 75.81 |
| CMF | 38.46 | N/A | N/A |
| RMGM | 26.45 | N/A | N/A |
| CD-SVD | 6.25 | 6.47 | 9.68 |

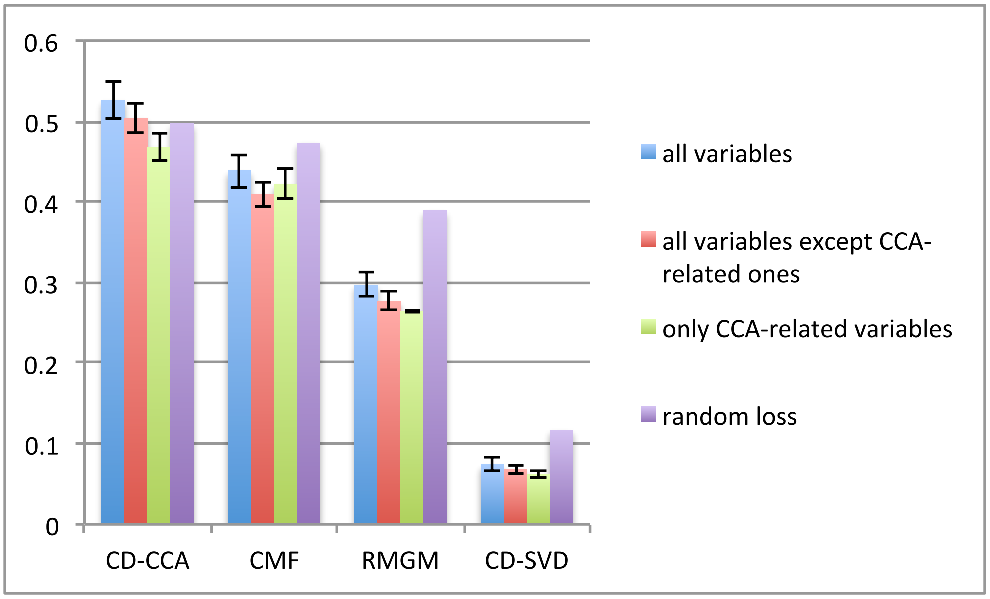


Figure 49: Classification loss in the mix of Supermarket and Yelp datasets

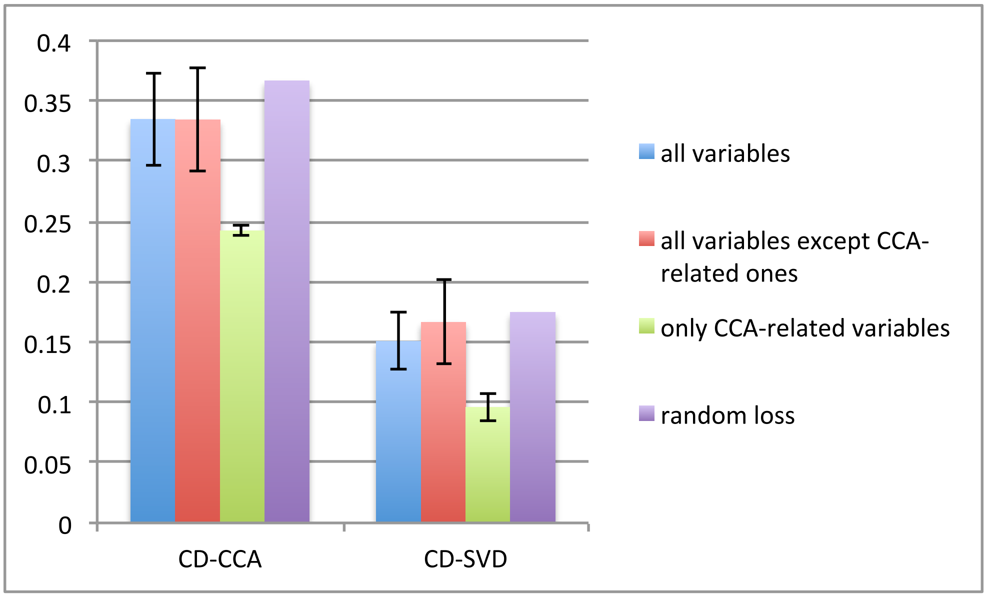


Figure 50: Classification loss in the mix of Supermarket and Imhonet dataset

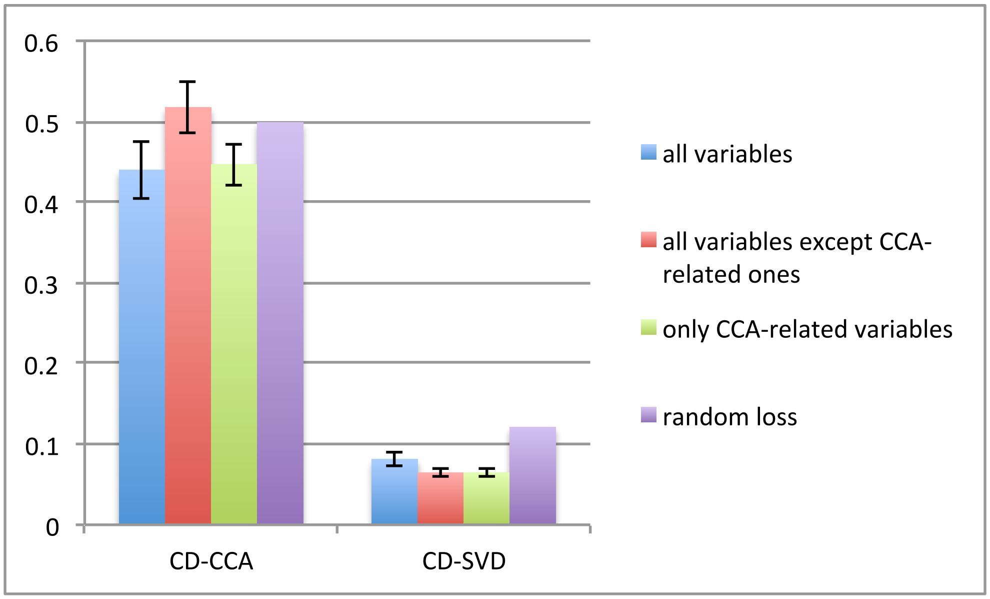


Figure 51: Classification loss in the mix of Yelp and Imhonet dataset

The results of the other two classifiers are comparable with the random baseline.

In Figure 51, we can see similar results for the mix of Yelp and Imhonet datasets. Again the classifier trained on CCA-related features is performing significantly better than the random baseline and better than the other two classifiers.

To summarize, we can see that the classifiers that are trained on the CCA-related features perform better in finding out the suitable domain pairs for cross-domain algorithms in the mixed datasets.

## 8.5 SUMMARY

In this chapter of the dissertation, we experimented on feasibility and generalizability of a domain-pair classifier, that can distinguish between suitable and non-suitable domain pairs for cross-domain recommendation. We have seen that the trained classifiers in general perform better than random baselines. We experimented on three different setups to research on within dataset, cross dataset, and mixed dataset classifiers. The results show that the trained classifiers can transfer the knowledge between datasets with very different natures and characteristics.

Also, we used three different sets of independent variables to examine the importance of CCA-related features in the classifiers. We noticed that in the within dataset experiments, adding CCA-related features does not boost the performance of domain pair classifiers. Using other dataset features can be as good as using the CCA-related features while working with one dataset. However, the classifiers trained on CCA-related features performed very well in mixed and cross dataset experiments. Thus, for the results to be generalizable beyond one dataset, we should rely on the CCA-related features.

Comparing the results of within dataset and cross dataset classifiers in the Supermarket and Yelp datasets, when trained on all features or all except CCA-related features, we can see that the loss of algorithms in the within dataset classifiers is less than the ones in the cross dataset classifiers. However, for the Imhonet dataset, the loss of within and cross datasets are very close to each other. This means that, if available, using domain pairs from the same dataset to train the classifier, provides us with a more accurate domain pair classifier. Also, for the classifiers trained on CCA-related features, the results of within and cross dataset classifiers are similar to each other. This leads us to using CCA-related features for training a classifier if the domain pair data from the same dataset is not available.

Eventually, since CCA-related features do not harm the within dataset classification results, and improve the mixed and cross dataset results, it would be better to keep them in the classifier.

Finally, to answer research question Q3, we conclude that having a domain pair classifier, although not perfect, is feasible and beneficial for cross-domain recommendation.

# 9.0 CONCLUSIONS AND DISCUSSION

In this chapter, we discuss the contributions of the proposed thesis, its limitations and delimitations, and the possible future work.

## 9.1 CONCLUSIONS

Our goals in this dissertation were to explore the added value of cross-domain recommendations in comparison with traditional within-domain recommendations, and to achieve some progress in uncovering the main mystery of cross-domain recommendation: how can we determine whether a pair of domains is a good candidate for applying cross-domain recommendation techniques?

To explore the cross-domain recommender systems' value, we proposed a cross-domain collaborative filtering approach, and its large-scale version, based on canonical correlation analysis in Chapter 3. To achieve the goals of this dissertation, we also designed the research questions Q1 to Q3 in Section 1.2.

The first question studies the feasibility and benefit of cross-domain recommender systems. The first part of this question (Q1.1) focuses on the reasons behind getting better results in cross-domain recommendations: if the better results are because of the additional data, better algorithms, or both of them. To answer this question, we performed a study as explained in Chapter 5. We examined the success and failure of our proposed approach, three other cross-domain collaborative filtering baselines, and a single-domain recommender system. We used three different datasets, introduced in Chapter 4, with various characteristics in the number of domains, nature of domains, and size of data to run the experiments.

We noticed different behavior of the algorithms in different datasets and discussed the potential reasons for these irregularities and their relationships to the design of cross-domain algorithms. For example, a potential reason for RMGM algorithm to perform very well in the Supermarket dataset and very poorly in the Yelp dataset is skewness of the Yelp dataset, compared to the Supermarket dataset. From these experiments, we concluded that both extra information and the approach that uses this extra information are important in achieving better results in cross-domain recommender systems. We have seen that CD-SVD algorithm sometimes perform better than SD-SVD algorithm, only because it has access to more information about the users from the source domain. Also, in many domain pairs in all of the datasets other cross-domain algorithms including our proposed CD-CCA has performed better than CD-SVD. All of these algorithms had both domains' data as their input, but they had different approaches to use this data.

Additionally, we observed that not all of the domain pairs in all of the datasets are suitable for cross-domain recommendations. For some of the domain pairs, especially in the Supermarket dataset, the single-domain recommendations can work better than the cross-domain ones.

We analyzed the time performance of algorithms and concluded that one of the strong points for the proposed CD-CCA is its time performance; it is the fastest cross-domain algorithm in average-sized datasets. In large-scale datasets CD-LCCA has a reasonably good time-performance.

The second part of first question (Q1.2) concentrate on the cross-domain collaborative filtering in the cold-start setting. Since one of the major contributions of the cross-domain collaborative filtering is assumed to be in the cases where there is too little information (ratings) available by users, we would like to understand how each of the cross-domain approaches perform in this setting, compared to the single-domain algorithm. The study designed in Chapter 6 aims to answer this question. In this chapter, we studied the trend of errors of algorithms based on the user profile sizes in both target and source domains. We examined this trend both by looking at the average errors over all of the domain pairs, and the error trend in each of the domain pairs of the three datasets. In some cases of the extreme cold-start situation CD-SVD and SD-SVD were not able to recommend any items to users. But, other cross-domain algorithms could recommend items with reasonable errors in these cases. We noticed an initial increase on average error, by target profile size increase in the cold-start setting that was attributed to the large number of domain pairs that had very small user profiles. Also, we have seen a decrease of error with larger target profile sizes in most of the cross-domain recommenders that was attributed to having more information about users and thus better recommendation results. We notices that RMGM algorithm performs very poorly in the extreme cold-start setting, when users have only one item in their target profile. We noticed that error of SD-SVD is surprisingly associated with source domain profile sizes and hypothesized that this phenomenon is because of a third factor correlated with both source domain profile size and error of SD-SVD. Also, we have seen that CD-CCA can use the external source information while avoiding the noise that comes with it. Based on these trends we concluded that CD-CCA can handle the cold-start problem better than the baseline cross-domain recommenders. Additionally, we inferred that the cross-domain algorithms can especially help the results in the cold-start setting.

Having answered the first research question, we studied the reasons behind the observed performance of cross-domain approaches by designing the experiments in Chapter 7. In this chapter we examined the three parts of our second research question: we studied canonical correlations between the domains, in addition to various dataset and domain-pair characteristics, as possible predictors of successful domain pairs. We concluded that, for each of the algorithms in each of the datasets, a different set of variables can have significant relationships with the performance of the algorithms. The size and direction of these relationships varied among the datasets and algorithms. The most consistent associations among the datasets and algorithms were CCA-related features and features related to KL-divergence. We discussed potential reasons for having such inconsistent or insignificant results, including few number of datapoints (domain pairs), only checking the linear associations or monotonic correlations, and different nature of datasets. We looked at domain-pairs with surprising and unexpected behaviors to understand if empirical behavior of domain pairs matches our intuitive expectations from them. We noticed some domain-pairs with high CCA and poor cross-domain performance, low CCA and good cross-domain results, and high CCA with good results that did not seem to have any intuitive relationship. We concluded that CCA, as a tool in detecting beneficial domain-pairs, can discover some counter-intuitive but useful relationships that lead to good cross-domain performance. However, it is better to rely on a combination of dataset characteristics and CCA of domain pairs to select the best-performing domains.

Finally, in Chapter 8.3, we answered the third research question (Q3): if we can classify the domain pairs into beneficial and non-beneficial ones for performing cross-domain recommendation. We designed a study that uses a SVM classifier to distinguish between the domain pairs with significantly better cross-domain recommendations, compared to the single-domain ones. We used three different sets of features (independent variables) to study the effect of CCA-related features in the classifiers. Also, we designed three setups for within dataset, cross dataset, and mixed dataset classification of domain pairs. In brief, we discovered that it is possible to classify the domain pairs into appropriate and inappropriate ones for cross-domain recommendations, with some error. We noticed that in most cases, adding the CCA-related features does not change the classification results within one dataset significantly. However, for having a transferable classifier among the datasets or a classifier on a mixed dataset, the CCA-related features work the best. Thus, we concluded that they play an important role in general for classifying the appropriate domain pairs for cross-domain collaborative filtering.

On the whole, we have concluded that:*a*) cross-domain collaborative filtering is useful in some domain pairs if the appropriate approach is selected to handle the extra information of the source domain; *b*) it helps to alleviate the cold-start problem (especially the proposed CD-CCA approach); *c*) some dataset-dependent data characteristics can define if a domain pair is a good selection for cross-domain recommendation or not; *d*) the CCA-related data characteristics are the most consistent ones across multiple datasets; *e*) however, it is better to rely on a mix of dataset features and CCA-related ones in figuring out the best domain pairs using correlation and regression analyses; and *f*) we can classify domain pairs into beneficial and non-beneficial domain pairs before performing the cross-domain recommendations[1] 1) relying on dataset descriptor data features for within dataset classifications; and 2) relying on CCA-related features for generalizable cross and mixed dataset classifications.

[1] with some classification loss that is less than random loss.

## 9.2 CONTRIBUTIONS

Although cross-domain recommender systems have been the focus of some researches recently, the past studies mostly focus on the feasibility of cross-domain recommenders and its benefits. In this thesis, we discovered that there are more complexities involved in performance and benefits of cross-domain recommenders: it may depend on the dataset, algorithm, and specifically choice of domain pairs.

To discover these complexities and the factors behind them, we proposed a cross-domain recommendation method based on canonical correlation analysis (CD-CCA) and its large-scale version (CD-LCCA) that can take into account the relatedness of domains while transferring the knowledge from one domain to the another. We tested the performance of this method with three baseline cross-domain and one single-domain recommenders on three different datasets. We explored the feasibility and benefit of cross-domain recommenders in the general, and cold-start settings. We performed correlation analysis and regression analysis on various dataset features to understand the cues that lead us in selecting the best domain pairs. We proposed CCA as the main data feature to distinguish between domain pairs. Finally, we proposed a domain pair classifier to distinguish between appropriate and not appropriate domain pairs, and studied the domain pairs with unexpected behaviors.

In summary, our contribution in this thesis are be as follows:

- To the best of our knowledge, we presented the first large scale study that examines the value of cross-domain recommendation approaches in a broad and diverse set of domain pairs.
- We designed a new cross-domain recommendation approach based on Canonical Correlation Analysis (CD-CCA) as well as the large-scale version of it (CD-LCCA).
- We conducted a study on the performance of this approach and baseline approaches to find out:
  - if the cross-domain recommendation results only improve because of added information, or if the recommendation algorithm also matters; and
  - if the cross-domain recommenders alleviate the cold-start problem.

- We designed a detailed set of experiments by correlation and regression analysis of dataset characteristics with the error and improvement of cross-domain approaches to find out:
  - the single-domain and domain-pair data characteristics that affect the prediction error of approaches;
  - the single-domain and cross-domain data characteristics that affect the amount of recommendation improvements; and
  - the nature of suitable domain pairs.
- We built a domain pair classifier that can distinguish between helpful and unhelpful domain pairs for each recommender algorithm.

Our contributions in this thesis will be useful in finding if a domain pair are suitable for cross-domain recommendations before performing the recommendations, finding the best auxiliary domains when alleviating the cold-start problem in a target domain, finding more reliable information about users of a domain, and performing cross-domain recommendations in datasets having shared users.

## 9.3 LIMITATIONS AND DELIMITATIONS

The limitations and delimitations of the proposed thesis are derived from both the data used for the studies, and the proposed approaches.

This study utilizes data from three resources, imposing several limitations to this research:

- All three datasets are sparse. Although it is natural in the domain of recommender systems to have sparse dataset, this sparsity might affect the results and our confidence in them.
- There are few domains in the Imhonet dataset compared to the other datasets. This results in less confident analysis for the relationship between data characteristics and recommendation results and less accurate predictions in the classifier.

- Although this dissertation is on cross-domain collaborative filtering recommender systems, two of the datasets are not based on classic ratings ("tastes") in the items they purchases. The Yelp dataset has user ratings, but is on business services rather then for individual products and services. The Supermarket dataset includes implicit "taste" of customers in their purchases, but it does not have an explicit preference feedback, nor the decision of user to rate an item or not. Only in the Imhonet dataset, we have an explicit rating of users on items, based on their taste.
- The Imhonet data was collected while there was a recommender system active in the system. Also, some of the users are guided to provide at least 20 ratings in the movies and books domains. This might cause a bias in the preference ratings of users and affect the analysis and recommendation results.
- The Supermarket dataset is gathered from purchases of people with special discounts in some domains (more specifically, healthy foods). This might cause a bias in the purchasing pattern of customers that can affect the analysis and recommendation results.

In addition, the following limitations and delimitations exist with regards to the approaches proposed in this thesis:

- While the data in recommender systems are sparse, CD-CCA assumes that it has access to all of the training records. To alleviate this problem, we use a common rule of thumb in recommender systems and fill in each of the unknown training records with an average rating calculated by average user rating, average item rating, and the global average of the data.
- CD-CCA is only able to capture the linear relationship between the domains. While there might be some non-linear relationship between the domains, CD-CCA is unable to capture it because of the linear nature of CCA. Kernel-CCA methods can capture the non-linear relationship between the domains. However, using kernel-CCA for recommendation requires a mapping from the source space to the kernel space and from the kernel space back to the target space, and performing the latter is not feasible in the extent of this thesis.
- The proposed approaches require shared user sets between the source and target domains.

- Because of the large size of Imhonet dataset and slow performance of RMGM and CMF algorithms we were not able to report their results in the dissertation.
- We only have four baseline approaches in the thesis. The results may not be generalizable to all recommendation approaches.
- The regression and Pearson correlation analyses of Chapter 7 rely on linear and monotonous associations among dependent and independent variables. As a result, we cannot find other possible associations in the data, including non-monotonic, polynomial, or exponential relationships.
- The analyses we performed in Chapter 7 lack enough power to make strong conclusions.
- Even after removing multicollinearity from the variables in Chapter 7, some of the regression models were not significant. This can be because of the few number of datapoints, the complicated nature of the dependent variables, or the hidden relationships between the dependent variables.
- Our approaches are not discovering any causal relationships between the dataset features and the recommender system results by performing the analyses.
- While other settings can be defined to classify between beneficial and non-beneficial domain pairs, the classifier presented in this thesis is a binary linear classifier distinguishing between significantly better performing cross-domain algorithms compared to SD-SVD. Thus, it does not capture possible non-linear relationships that may exist in the data.
- All of the experiments are on offline datasets and may have different results in the wild.
- This thesis is focused on empirical aspects of cross-domain collaborative filtering and does not analyze the effects of adding external source domain data theoretically. While having empirical evidence for the benefits and disadvantages of cross-domain recommender systems is valuable and needed, it is important to support this evidence with theoretical definitions and proofs. It can provide a strong foundation for the empirical results that we have seen in this dissertation and explain them.

## 9.4 FUTURE WORK

Popularity of cross-domain recommender systems is increasing because of their promises for alleviating problems such as cold-start and sparsity. However, there are many aspects of these type of recommender systems, such as domain pair selection, intelligent transfer of information from the source domain to the target domain, domain definition, and large-scale algorithms, that need more research. Although we have done an extensive study on domain pair selection in this thesis, the definition of domains are coming from predefined, manual, categorization of items. One of the research questions that can be addressed in continuation of this thesis is what makes a domain definition to be good. Especially in datasets similar to the Supermarket dataset, the item domains can be very similar to each other. Defining metrics to define a good domain and automatic ways to separate the domains based on these metrics can be an interesting extension of this work.

In this thesis, our focus was on finding out the data characteristics that can lead to a good domain pair and the feasibility of domain pair classification for the cross-domain recommendation. Study on different classifier approaches in different setups or nonlinear association analysis may provide a better idea on finding the best domain pairs for a dataset.

In addition to the empirical studies reported in this thesis, a future direction can take a theoretical view to the problem to analyze the effect of adding extra source domain information, with different dataset characteristics, to the recommendation results. As we have discussed in the limitations section, a theoretical analysis can strengthen the empirical results and explain the irregularities in them. Given the limited literature on this aspect of cross-domain recommenders, such work can be very valuable.

Our proposed approaches (CD-CCA and CD-LCCA) work on a complete matrix of ratings. As another possible future work in this thesis, a sparse version of these algorithms that relies on the observed user ratings only, and thus an approximation of CCA, can be developed. Similarly, an online version of these algorithms can be developed to be deployed in a real world setting. In this case, one can experiment on the real-time recommender system and extend the evaluations with A/B testing. Another possible extension for the proposed algorithms is adopting them for the implicit feedback data. Here, we transformed the implicit feedback in the Supermarket dataset to a similar rating scale. However, we can design an algorithm that models the implicit feedback directly.

Another focus of this dissertation was on pairs of domains. An interesting research direction can be studying the effect of multiple auxiliary domains on the target domain recommendations, either having a CCA-based algorithm or other transfer learning approaches. On of the challenges for this research would be designing an evaluation framework for it. Since the number of different combinations for a set of source domains is exponential, evaluating the best selection of domains need some heuristics to reduce the search space for this problem.

Finally, we have only researched on collaborative filtering cross-domain approaches. However, there are many other possible resources, such as texts, tags, and context, that can be used in cross-domain recommendations. Including these resources in the cross-domain recommender systems may both boost their performance and increase their interpretability. For example, we may find out some domain similarities by analyzing the texts associated with two domains. In a related note, we have used the word "semantics" for two heuristically similar domains. Using the text associated with domains we can check if there are any semantics that can be discovered between such domains.

# APPENDIX A

# SUPERMARKET DATA FIGURES AND TABLES

## A.1 SUPERMARKET DATA DOMAIN MAPPING

Table 36: Mapping of supermarket item categories to domains

| | | |
|---|---|---|
| alcoholic drinks & cigarrettes | spirits | 28 |
| alcoholic drinks & cigarrettes | beer | 34 |
| alcoholic drinks & cigarrettes | wine | 110 |
| alcoholic drinks & cigarrettes | cigarettes/tobacco | 1502 |
| beauty | soaps & body wash | 8580 |
| beauty | skin care | 5800 |
| beauty | mens toiletries | 5042 |
| beauty | health & beauty gift | 78 |
| beauty | hair care | 10213 |
| beauty | hair accessories | 15 |
| beauty | facial tissues | 7368 |
| beauty | deodorants | 6988 |
| beauty | cosmetics/toiletries | 2009 |
| beauty | beauty/trial travel | 2278 |
| bread | instore bread | 24174 |
| bread | bread rolls & fbread | 8586 |
| bread | bh bought in easter | 29 |
| bread | bakery bought in | 26599 |
| canned & pickled | pickled vegetables | 3332 |
| canned & pickled | meal bases | 14939 |
| canned & pickled | canned veg | 16026 |
| canned & pickled | canned meals | 4961 |

## Mapping of supermarket item categories to domains

| parent category (domain) | subcategory | number of records in subcategory |
|---|---|---|
| canned & pickled | antip/olive/dip/pate | 6441 |
| clothing | seasonal apparel | 358 |
| clothing | mix womenswear | 2489 |
| clothing | mix menswear | 288 |
| clothing | mix girlswear 2-6 | 463 |
| clothing | mix boyswear 2-6 | 293 |
| clothing | mix babywear | 414 |
| clothing | menswear | 1 |
| clothing | ladieswear | 2 |
| clothing | hosiery | 1708 |
| clothing | girlswear | 1 |
| clothing | family underwear | 731 |
| clothing | family socks | 1324 |
| clothing | boyswear | 1 |
| clothing | babywear | 386 |
| cooking essentials | vinegar | 1499 |
| cooking essentials | sugar/sweeteners | 5759 |
| cooking essentials | spices/herbs | 8086 |
| cooking essentials | sauces/relish | 10527 |
| cooking essentials | salad dressings | 4278 |
| cooking essentials | rice | 6409 |
| cooking essentials | pasta | 12226 |
| cooking essentials | oils | 4683 |
| cooking essentials | flour | 3820 |
| cooking essentials | cooking | 14359 |
| dairy | spec/fresh cheese | 3416 |
| dairy | grocery milk | 9335 |
| dairy | gourmet cheese | 8353 |
| dairy | dy milk | 46107 |
| dairy | chilled spreads | 13225 |
| dairy | cheese dairy | 24851 |
| discounts & coupons | select ctomr discnt | 48 |
| discounts & coupons | dummy do not touch | 23 |
| discounts & coupons | Supermarket mcard disc | 41 |
| discounts & coupons | Supermarket insurance disc | 170 |
| discounts & coupons | Supermarket finsvc $10 off | 9293 |
| discounts & coupons | ancil services | 10929 |
| events | seasonal events | 1479 |
| events | party goods | 1637 |
| events | events | 1479 |

## Mapping of supermarket item categories to domains

| parent category (domain) | subcategory | number of records in subcategory |
|---|---|---|
| events | easter | 3222 |
| events | christmas gen merch | 11 |
| fish, meat, poultry & eggs | smallgoods dairy | 12177 |
| fish, meat, poultry & eggs | smallgoods | 5880 |
| fish, meat, poultry & eggs | sliced meats | 14091 |
| fish, meat, poultry & eggs | seafood (mt) | 2134 |
| fish, meat, poultry & eggs | seafood (dl) | 4896 |
| fish, meat, poultry & eggs | sausages | 9980 |
| fish, meat, poultry & eggs | poultry-frozen | 53 |
| fish, meat, poultry & eggs | poultry (mt) | 18592 |
| fish, meat, poultry & eggs | poultry (dl) | 4575 |
| fish, meat, poultry & eggs | pork | 4983 |
| fish, meat, poultry & eggs | lamb | 6927 |
| fish, meat, poultry & eggs | hams/bacon | 771 |
| fish, meat, poultry & eggs | game | 462 |
| fish, meat, poultry & eggs | frozen meat | 4 |
| fish, meat, poultry & eggs | fish - dairy | 2212 |
| fish, meat, poultry & eggs | eggs | 13810 |
| fish, meat, poultry & eggs | continental | 5438 |
| fish, meat, poultry & eggs | canned fish | 15551 |
| fish, meat, poultry & eggs | beef | 20583 |
| fish, meat, poultry & eggs | bbq | 5889 |
| fruit & vegetables | soft vegetables | 111485 |
| fruit & vegetables | organic fruit & veg | 1013 |
| fruit & vegetables | hard veg & mushroom | 62138 |
| fruit & vegetables | garden greens | 1 |
| fruit & vegetables | fruit-shelf stable | 6110 |
| fruit & vegetables | fruit snacks | 43198 |
| fruit & vegetables | fruit desserts | 70676 |
| fruit & vegetables | frozen vegetables | 14419 |
| gifts | gift cards | 518 |
| gifts | floral | 1487 |
| gifts | christmas gr non fd | 576 |
| gifts | cards/wraps | 3108 |
| gifts | 3rd party giftcard | 627 |
| health | vitamins | 2165 |
| health | sanitary protection | 5837 |
| health | medicinal products | 8036 |
| health | infant personal | 3172 |
| health | infant nappies | 2839 |

## Mapping of supermarket item categories to domains

| parent category (domain) | subcategory | number of records in subcategory |
|---|---|---|
| health | first aid | 48 |
| health | dental health | 9780 |
| home cleaning | toilet paper | 8736 |
| home cleaning | paper towels | 4394 |
| home cleaning | laundry accessories | 5373 |
| home cleaning | laundry | 4545 |
| home cleaning | household gloves | 967 |
| home cleaning | dishwashing | 6333 |
| home cleaning | cleaning goods | 13062 |
| home cleaning | brushware | 1020 |
| home cleaning | aircare & disinfect | 3009 |
| home indoor | stationery | 2614 |
| home indoor | shopping bags | 849 |
| home indoor | shoe care | 922 |
| home indoor | photographics | 62 |
| home indoor | nursery | 7 |
| home indoor | kitchenware | 4810 |
| home indoor | kitchen needs/bags | 10317 |
| home indoor | household appliances | 142 |
| home indoor | homewares | 1304 |
| home indoor | home textiles | 385 |
| home indoor | home organisation | 55 |
| home indoor | heating & cooling | 82 |
| home indoor | hardware | 658 |
| home indoor | electrical | 2734 |
| home indoor | audio / video | 160 |
| home outdoor | pool and outdoor acc | 40 |
| home outdoor | picnic pool bbq acc | 1321 |
| home outdoor | outdoor living | 12 |
| home outdoor | leisure | 271 |
| home outdoor | garden non greens | 700 |
| home outdoor | disposable tableware | 4166 |
| home outdoor | auto | 311 |
| international food | mexican foods | 7011 |
| international food | international foods | 2540 |
| international food | instant noodles | 4339 |
| international food | asian & indian foods | 10340 |
| leisure | toys & hobbies | 1291 |
| leisure | telco | 1016 |
| leisure | prerecorded media | 776 |

## Mapping of supermarket item categories to domains

| parent category (domain) | subcategory | number of records in subcategory |
|---|---|---|
| leisure | magazines | 7869 |
| leisure | books | 771 |
| pets | pet food | 25451 |
| pets | pet accessories | 746 |
| pets | pest control | 1662 |
| pets | fresh pet food | 2488 |
| pets | bird food | 738 |
| prepared meals & snacks | sushi | 166 |
| prepared meals & snacks | soup | 8364 |
| prepared meals & snacks | snacks | 38451 |
| prepared meals & snacks | salads | 1606 |
| prepared meals & snacks | salad bar | 19 |
| prepared meals & snacks | protein & meals | 2479 |
| prepared meals & snacks | prepared foods | 6 |
| prepared meals & snacks | packaged salads | 27877 |
| prepared meals & snacks | nuts/dried | 9142 |
| prepared meals & snacks | nutritional snacks | 12093 |
| prepared meals & snacks | meals | 12809 |
| prepared meals & snacks | local foods | 281 |
| prepared meals & snacks | instore pseudo brand | 2 |
| prepared meals & snacks | infant food/formula | 6237 |
| prepared meals & snacks | hmr | 371 |
| prepared meals & snacks | heat & eat frozen | 10533 |
| prepared meals & snacks | healthfoods | 1288 |
| prepared meals & snacks | health foods | 15613 |
| prepared meals & snacks | grab & go | 724 |
| prepared meals & snacks | frozen snacks | 1505 |
| prepared meals & snacks | entertainment | 11761 |
| prepared meals & snacks | dried fruit/nuts | 6905 |
| prepared meals & snacks | dl prepared foods | 709 |
| prepared meals & snacks | dl hot pies & foods | 97 |
| prepared meals & snacks | convenience meals | 2435 |
| prepared meals & snacks | convenience frozen | 12357 |
| prepared meals & snacks | christmas gr food | 1818 |
| prepared meals & snacks | cereal | 21100 |
| prepared meals & snacks | biscuits & cookies | 49462 |
| prepared meals & snacks | bars gum pocket pack | 21212 |
| soft drinks, tea & coffee | tea | 5664 |
| soft drinks, tea & coffee | still water | 8545 |
| soft drinks, tea & coffee | softdrinks | 38018 |

## Mapping of supermarket item categories to domains

| parent category (domain) | subcategory | number of records in subcategory |
|---|---|---|
| soft drinks, tea & coffee | non alco wine & hbrw | 510 |
| soft drinks, tea & coffee | milk additives | 2238 |
| soft drinks, tea & coffee | juices/drinks | 9608 |
| soft drinks, tea & coffee | juices & cordials | 6589 |
| soft drinks, tea & coffee | juices | 11155 |
| soft drinks, tea & coffee | instore cafe | 265 |
| soft drinks, tea & coffee | energy/sport/icedtea | 9004 |
| soft drinks, tea & coffee | cordial | 1699 |
| soft drinks, tea & coffee | coffee | 5654 |
| sweets | spreads | 9258 |
| sweets | patisserie | 130 |
| sweets | instore cake | 11276 |
| sweets | ice cream | 17393 |
| sweets | frozen desserts | 1381 |
| sweets | desserts (gr) | 5285 |
| sweets | desserts | 5683 |
| sweets | confectionery | 34541 |
| sweets | christmas confect | 87 |
| sweets | chilled desserts | 47169 |
| sweets | brought in seasonal | 1293 |
| sweets | boxed chocolates | 3870 |
| sweets | baking mixes | 3021 |
| sweets | bakery snacks | 5854 |
| sweets | bakery packaged cake | 6668 |
| sweets | bake instore seasonl | 1680 |

## A.2 DOMAIN PAIR STATISTICS FOR THE SUPERMARKET DATASET

Table 37: Domain and domain-pair data size statistics for the supermarket dataset

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| bread | canned & pickled | 1434 | 608 | 929 | 0.0290 | 0.0179 | 2.3586 | 1.5436 | 0.6545 | 1.6200 |
| bread | dairy | 1498 | 612 | 1149 | 0.0278 | 0.0209 | 2.4477 | 1.3037 | 0.5326 | 1.3325 |
| bread | events | 991 | 588 | 859 | 0.0344 | 0.0064 | 1.6854 | 1.1537 | 0.6845 | 5.4176 |
| bread | fruit & vegetables | 1504 | 612 | 980 | 0.0277 | 0.0642 | 2.4575 | 1.5347 | 0.6245 | 0.4323 |
| bread | gifts | 1111 | 594 | 573 | 0.0325 | 0.0083 | 1.8704 | 1.9389 | 1.0366 | 3.9188 |
| bread | home cleaning | 1437 | 612 | 1123 | 0.0287 | 0.0185 | 2.3480 | 1.2796 | 0.5450 | 1.5541 |
| bread | home outdoor | 1096 | 597 | 518 | 0.0328 | 0.0098 | 1.8358 | 2.1158 | 1.1525 | 3.3378 |
| bread | international food | 1342 | 610 | 1184 | 0.0299 | 0.0095 | 2.2000 | 1.1334 | 0.5152 | 3.1548 |
| canned & pickled | bread | 1434 | 929 | 608 | 0.0179 | 0.0290 | 1.5436 | 2.3586 | 1.5280 | 0.6173 |
| canned & pickled | dairy | 1440 | 929 | 1147 | 0.0179 | 0.0216 | 1.5501 | 1.2554 | 0.8099 | 0.8259 |
| canned & pickled | events | 973 | 921 | 859 | 0.0210 | 0.0064 | 1.0565 | 1.1327 | 1.0722 | 3.2627 |
| canned & pickled | fruit & vegetables | 1446 | 929 | 980 | 0.0178 | 0.0664 | 1.5565 | 1.4755 | 0.9480 | 0.2680 |
| canned & pickled | gifts | 1089 | 919 | 571 | 0.0204 | 0.0084 | 1.1850 | 1.9072 | 1.6095 | 2.4290 |
| canned & pickled | home cleaning | 1397 | 929 | 1123 | 0.0182 | 0.0189 | 1.5038 | 1.2440 | 0.8272 | 0.9641 |
| canned & pickled | home outdoor | 1071 | 927 | 516 | 0.0203 | 0.0100 | 1.1553 | 2.0756 | 1.7965 | 2.0333 |
| canned & pickled | international food | 1319 | 928 | 1182 | 0.0190 | 0.0096 | 1.4213 | 1.1159 | 0.7851 | 1.9723 |
| dairy | bread | 1498 | 1149 | 612 | 0.0209 | 0.0278 | 1.3037 | 2.4477 | 1.8775 | 0.7505 |
| dairy | canned & pickled | 1440 | 1147 | 929 | 0.0216 | 0.0179 | 1.2554 | 1.5501 | 1.2347 | 1.2108 |
| dairy | fruit & vegetables | 1520 | 1150 | 980 | 0.0207 | 0.0637 | 1.3217 | 1.5510 | 1.1735 | 0.3241 |
| dairy | home cleaning | 1448 | 1147 | 1123 | 0.0215 | 0.0184 | 1.2624 | 1.2894 | 1.0214 | 1.1688 |
| dairy | international food | 1351 | 1144 | 1184 | 0.0224 | 0.0094 | 1.1809 | 1.1410 | 0.9662 | 2.3796 |
| events | bread | 991 | 859 | 588 | 0.0064 | 0.0344 | 1.1537 | 1.6854 | 1.4609 | 0.1846 |
| events | canned & pickled | 973 | 859 | 921 | 0.0064 | 0.0210 | 1.1327 | 1.0565 | 0.9327 | 0.3065 |
| events | fruit & vegetables | 997 | 859 | 973 | 0.0063 | 0.0747 | 1.1607 | 1.0247 | 0.8828 | 0.0848 |
| fruit & vegetables | bread | 1504 | 980 | 612 | 0.0642 | 0.0277 | 1.5347 | 2.4575 | 1.6013 | 2.3131 |
| fruit & vegetables | canned & pickled | 1446 | 980 | 929 | 0.0664 | 0.0178 | 1.4755 | 1.5565 | 1.0549 | 3.7320 |
| fruit & vegetables | dairy | 1520 | 980 | 1150 | 0.0637 | 0.0207 | 1.5510 | 1.3217 | 0.8522 | 3.0857 |
| fruit & vegetables | events | 997 | 973 | 859 | 0.0747 | 0.0063 | 1.0247 | 1.1607 | 1.1327 | 11.7895 |
| fruit & vegetables | gifts | 1117 | 973 | 573 | 0.0734 | 0.0083 | 1.1480 | 1.9494 | 1.6981 | 8.8739 |
| fruit & vegetables | home cleaning | 1452 | 979 | 1123 | 0.0661 | 0.0183 | 1.4831 | 1.2930 | 0.8718 | 3.6005 |
| fruit & vegetables | home outdoor | 1097 | 978 | 518 | 0.0734 | 0.0098 | 1.1217 | 2.1178 | 1.8880 | 7.4688 |
| fruit & vegetables | international food | 1353 | 980 | 1185 | 0.0688 | 0.0094 | 1.3806 | 1.1418 | 0.8270 | 7.3155 |
| gifts | bread | 1111 | 573 | 594 | 0.0083 | 0.0325 | 1.9389 | 1.8704 | 0.9646 | 0.2552 |
| gifts | canned & pickled | 1089 | 571 | 919 | 0.0084 | 0.0204 | 1.9072 | 1.1850 | 0.6213 | 0.4117 |
| gifts | fruit & vegetables | 1117 | 573 | 973 | 0.0083 | 0.0734 | 1.9494 | 1.1480 | 0.5889 | 0.1127 |
| gifts | home outdoor | 902 | 547 | 502 | 0.0093 | 0.0108 | 1.6490 | 1.7968 | 1.0896 | 0.8622 |
| home cleaning | bread | 1437 | 1123 | 612 | 0.0185 | 0.0287 | 1.2796 | 2.3480 | 1.8350 | 0.6435 |

Domain and domain-pair data size statistics for the supermarket dataset contd.

| source | target | user size | source item size | target item size | source density | target density | user to source item ratio | user to target item ratio | source to target item ratio | source to target density ratio |
|---|---|---|---|---|---|---|---|---|---|---|
| home cleaning | canned & pickled | 1397 | 1123 | 929 | 0.0189 | 0.0182 | 1.2440 | 1.5038 | 1.2088 | 1.0372 |
| home cleaning | dairy | 1448 | 1123 | 1147 | 0.0184 | 0.0215 | 1.2894 | 1.2624 | 0.9791 | 0.8556 |
| home cleaning | fruit & vegetables | 1452 | 1123 | 979 | 0.0183 | 0.0661 | 1.2930 | 1.4831 | 1.1471 | 0.2777 |
| home cleaning | international food | 1312 | 1123 | 1184 | 0.0196 | 0.0096 | 1.1683 | 1.1081 | 0.9485 | 2.0329 |
| home outdoor | bread | 1096 | 518 | 597 | 0.0098 | 0.0328 | 2.1158 | 1.8358 | 0.8677 | 0.2996 |
| home outdoor | canned & pickled | 1071 | 516 | 927 | 0.0100 | 0.0203 | 2.0756 | 1.1553 | 0.5566 | 0.4918 |
| home outdoor | fruit & vegetables | 1097 | 518 | 978 | 0.0098 | 0.0734 | 2.1178 | 1.1217 | 0.5297 | 0.1339 |
| home outdoor | gifts | 902 | 502 | 547 | 0.0108 | 0.0093 | 1.7968 | 1.6490 | 0.9177 | 1.1599 |
| international food | bread | 1342 | 1184 | 610 | 0.0095 | 0.0299 | 1.1334 | 2.2000 | 1.9410 | 0.3170 |
| international food | canned & pickled | 1319 | 1182 | 928 | 0.0096 | 0.0190 | 1.1159 | 1.4213 | 1.2737 | 0.5070 |
| international food | dairy | 1351 | 1184 | 1144 | 0.0094 | 0.0224 | 1.1410 | 1.1809 | 1.0350 | 0.4202 |
| international food | fruit & vegetables | 1353 | 1185 | 980 | 0.0094 | 0.0688 | 1.1418 | 1.3806 | 1.2092 | 0.1367 |
| international food | home cleaning | 1312 | 1184 | 1123 | 0.0096 | 0.0196 | 1.1081 | 1.1683 | 1.0543 | 0.4919 |

Table 38: Each domain ratings' central tendency and dispersion statistics for the supermarket dataset

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| bread | canned & pickled | 4.6617 | 3.8064 | 3.5065 | 2.9594 | 1.6692 | 2.0968 | 23.7129 | 9.2576 |
| bread | dairy | 4.6474 | 5.8235 | 3.4930 | 3.7863 | 1.6763 | 2.0883 | 23.5075 | 53.1673 |
| bread | events | 4.8059 | 2.3285 | 3.6030 | 1.9607 | 2.0971 | 1.8357 | 25.9102 | 1.9093 |
| bread | fruit & vegetables | 4.6458 | 6.4902 | 3.4917 | 4.5464 | 1.6771 | 15.0120 | 23.4881 | 46.3169 |
| bread | gifts | 4.7498 | 2.3800 | 3.5572 | 2.0322 | 2.1251 | 1.8100 | 25.1426 | 7.8743 |
| bread | home cleaning | 4.6724 | 3.1655 | 3.5130 | 2.5181 | 1.6638 | 1.4172 | 23.7344 | 4.8737 |
| bread | home outdoor | 4.7619 | 2.4366 | 3.5803 | 2.0544 | 1.6190 | 1.7817 | 25.1580 | 1.9147 |
| bread | international food | 4.6774 | 3.2257 | 3.5126 | 2.6125 | 2.1613 | 1.3871 | 23.9487 | 5.0982 |
| canned & pickled | bread | 3.8064 | 4.6617 | 2.9594 | 3.5065 | 2.0968 | 1.6692 | 9.2576 | 23.7129 |
| canned & pickled | dairy | 3.8194 | 5.8468 | 2.9606 | 3.7990 | 2.0903 | 2.0766 | 9.5062 | 53.7226 |
| canned & pickled | events | 3.9364 | 2.3299 | 3.0366 | 1.9601 | 2.0318 | 1.8351 | 10.4737 | 1.9292 |
| canned & pickled | fruit & vegetables | 3.8190 | 6.5327 | 2.9597 | 4.5595 | 2.0905 | 15.4050 | 9.5035 | 47.7325 |
| canned & pickled | gifts | 3.8738 | 2.3657 | 3.0025 | 2.0354 | 2.0631 | 1.8171 | 9.7034 | 7.7832 |
| canned & pickled | home cleaning | 3.8307 | 3.1729 | 2.9693 | 2.5244 | 2.0847 | 1.4135 | 9.5770 | 4.9006 |
| canned & pickled | home outdoor | 3.8907 | 2.4376 | 3.0096 | 2.0539 | 2.0546 | 1.7812 | 10.0123 | 1.9308 |
| canned & pickled | international food | 3.8396 | 3.2289 | 2.9767 | 2.6163 | 2.0802 | 1.3856 | 9.5532 | 5.1027 |
| dairy | bread | 5.8235 | 4.6474 | 3.7863 | 3.4930 | 2.0883 | 1.6763 | 53.1673 | 23.5075 |
| dairy | canned & pickled | 5.8468 | 3.8194 | 3.7990 | 2.9606 | 2.0766 | 2.0903 | 53.7226 | 9.5062 |
| dairy | fruit & vegetables | 5.8236 | 6.5023 | 3.7873 | 4.5447 | 2.0882 | 15.0480 | 53.2220 | 47.1214 |
| dairy | home cleaning | 5.8574 | 3.1666 | 3.7975 | 2.5196 | 2.0713 | 1.4167 | 53.9368 | 4.8671 |
| dairy | international food | 5.8727 | 3.2249 | 3.7947 | 2.6111 | 1.0636 | 1.3876 | 54.6240 | 5.0961 |
| events | bread | 2.3285 | 4.8059 | 1.9607 | 3.6030 | 1.8357 | 2.0971 | 1.9093 | 25.9102 |
| events | canned & pickled | 2.3299 | 3.9364 | 1.9601 | 3.0366 | 1.8351 | 2.0318 | 1.9292 | 10.4737 |
| events | fruit & vegetables | 2.3274 | 6.8562 | 1.9613 | 4.7089 | 1.8363 | 3.5719 | 1.9028 | 55.1503 |
| fruit & vegetables | bread | 6.4902 | 4.6458 | 4.5464 | 3.4917 | 15.0120 | 1.6771 | 46.3169 | 23.4881 |
| fruit & vegetables | canned & pickled | 6.5327 | 3.8190 | 4.5595 | 2.9597 | 15.4050 | 2.0905 | 47.7325 | 9.5035 |
| fruit & vegetables | dairy | 6.5023 | 5.8236 | 4.5447 | 3.7873 | 15.0480 | 2.0882 | 47.1214 | 53.2220 |
| fruit & vegetables | events | 6.8562 | 2.3274 | 4.7089 | 1.9613 | 3.5719 | 1.8363 | 55.1503 | 1.9028 |
| fruit & vegetables | gifts | 6.7557 | 2.3794 | 4.6663 | 2.0325 | 1.6221 | 1.8103 | 52.0547 | 7.8622 |
| fruit & vegetables | home cleaning | 6.5364 | 3.1659 | 4.5637 | 2.5186 | 15.3260 | 1.4170 | 47.7540 | 4.8647 |
| fruit & vegetables | home outdoor | 6.7654 | 2.4358 | 4.6665 | 2.0548 | 2.7840 | 1.7821 | 52.8557 | 1.9126 |
| fruit & vegetables | international food | 6.5908 | 3.2246 | 4.5828 | 2.6119 | 2.1880 | 1.3877 | 48.9024 | 5.0948 |
| gifts | bread | 2.3800 | 4.7498 | 2.0322 | 3.5572 | 1.8100 | 2.1251 | 7.8743 | 25.1426 |
| gifts | canned & pickled | 2.3657 | 3.8738 | 2.0354 | 3.0025 | 1.8171 | 2.0631 | 7.7832 | 9.7034 |
| gifts | fruit & vegetables | 2.3794 | 6.7557 | 2.0325 | 4.6663 | 1.8103 | 1.6221 | 7.8622 | 52.0547 |
| gifts | home outdoor | 2.3977 | 2.4335 | 2.0234 | 2.0552 | 1.8012 | 1.7832 | 8.9933 | 1.8534 |
| home cleaning | bread | 3.1655 | 4.6724 | 2.5181 | 3.5130 | 1.4172 | 1.6638 | 4.8737 | 23.7344 |
| home cleaning | canned & pickled | 3.1729 | 3.8307 | 2.5244 | 2.9693 | 1.4135 | 2.0847 | 4.9006 | 9.5770 |
| home cleaning | dairy | 3.1666 | 5.8574 | 2.5196 | 3.7975 | 1.4167 | 2.0713 | 4.8671 | 53.9368 |

Each domain ratings' central tendency and dispersion statistics for the supermarket dataset contd.

| source | target | source mean rating | target mean rating | source median rating | target median rating | source mode rating | target mode rating | source var. rating | target var. rating |
|---|---|---|---|---|---|---|---|---|---|
| home cleaning | fruit & vegetables | 3.1659 | 6.5364 | 2.5186 | 4.5637 | 1.4170 | 15.3260 | 4.8647 | 47.7540 |
| home cleaning | international food | 3.1822 | 3.2354 | 2.5298 | 2.6223 | 1.4089 | 1.3823 | 4.9562 | 5.1364 |
| home outdoor | bread | 2.4366 | 4.7619 | 2.0544 | 3.5803 | 1.7817 | 1.6190 | 1.9147 | 25.1580 |
| home outdoor | canned & pickled | 2.4376 | 3.8907 | 2.0539 | 3.0096 | 1.7812 | 2.0546 | 1.9308 | 10.0123 |
| home outdoor | fruit & vegetables | 2.4358 | 6.7654 | 2.0548 | 4.6665 | 1.7821 | 2.7840 | 1.9126 | 52.8557 |
| home outdoor | gifts | 2.4335 | 2.3977 | 2.0552 | 2.0234 | 1.7832 | 1.8012 | 1.8534 | 8.9933 |
| international food | bread | 3.2257 | 4.6774 | 2.6125 | 3.5126 | 1.3871 | 2.1613 | 5.0982 | 23.9487 |
| international food | canned & pickled | 3.2289 | 3.8396 | 2.6163 | 2.9767 | 1.3856 | 2.0802 | 5.1027 | 9.5532 |
| international food | dairy | 3.2249 | 5.8727 | 2.6111 | 3.7947 | 1.3876 | 1.0636 | 5.0961 | 54.6240 |
| international food | fruit & vegetables | 3.2246 | 6.5908 | 2.6119 | 4.5828 | 1.3877 | 2.1880 | 5.0948 | 48.9024 |
| international food | home cleaning | 3.2354 | 3.1822 | 2.6223 | 2.5298 | 1.3823 | 1.4089 | 5.1364 | 4.9562 |

Table 39: Each domain and domain-pair ratings' dispersion statistics for the supermarket dataset

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| bread | canned & pickled | 0.7974 | 20.1432 | 19.5878 | 111.7299 | 131.0856 | 35.3695 | 7.8870 | 4.5297 |
| bread | dairy | 0.0376 | 12.3587 | 9.2668 | 84.7521 | 132.0583 | 89.5021 | 7.9137 | 6.4620 |
| bread | events | 1.2489 | 20.7659 | 23.6158 | 268.2549 | 128.8670 | 158.3910 | 7.8300 | 9.9263 |
| bread | fruit & vegetables | 0.6232 | 18.7729 | 17.8555 | 62.3595 | 132.1253 | 33.1809 | 7.9163 | 4.2681 |
| bread | gifts | 1.9945 | 20.6689 | 28.4264 | 316.3883 | 134.1996 | 3589.9013 | 8.0206 | 55.1349 |
| bread | home cleaning | 0.0920 | 13.7158 | 11.1967 | 103.3222 | 130.9721 | 75.6465 | 7.8811 | 6.0813 |
| bread | home outdoor | 0.6154 | 17.3764 | 15.0650 | 225.0669 | 134.3117 | 166.3014 | 8.0474 | 9.7343 |
| bread | international food | 0.2358 | 15.1743 | 12.9731 | 134.7176 | 132.2087 | 64.4583 | 7.9308 | 5.7741 |
| canned & pickled | bread | 0.4832 | 10.8096 | 2.0752 | 182.4986 | 35.3695 | 131.0856 | 4.5297 | 7.8870 |
| canned & pickled | dairy | 0.3864 | 10.9661 | 3.9712 | 151.8759 | 36.3366 | 88.2763 | 4.5936 | 6.4264 |
| canned & pickled | events | 1.1012 | 14.5539 | 2.3273 | 285.8555 | 34.2207 | 156.9432 | 4.4653 | 9.8891 |
| canned & pickled | fruit & vegetables | 0.0300 | 8.8319 | 5.2962 | 74.7694 | 36.3406 | 35.2655 | 4.5938 | 4.3636 |
| canned & pickled | gifts | 1.6880 | 15.0459 | 1.8458 | 328.3858 | 35.2596 | 3716.3020 | 4.5125 | 56.6101 |
| canned & pickled | home cleaning | 0.2919 | 9.8128 | 2.7405 | 147.2051 | 36.0978 | 75.0585 | 4.5789 | 6.0571 |
| canned & pickled | home outdoor | 0.8705 | 13.3895 | 2.0149 | 263.4571 | 35.5754 | 165.3755 | 4.5391 | 9.7181 |
| canned & pickled | international food | 0.1529 | 9.6909 | 2.6327 | 156.7528 | 35.3256 | 64.5412 | 4.5306 | 5.7773 |
| dairy | bread | 0.0265 | 11.0937 | 7.9055 | 99.0383 | 89.5021 | 132.0583 | 6.4620 | 7.9137 |
| dairy | canned & pickled | 0.5985 | 18.8501 | 17.5346 | 104.9710 | 88.2763 | 36.3366 | 6.4264 | 4.5936 |
| dairy | fruit & vegetables | 0.3525 | 16.0073 | 14.5840 | 57.1452 | 89.0897 | 35.4211 | 6.4505 | 4.3693 |
| dairy | home cleaning | 0.0469 | 13.3472 | 10.9040 | 100.6738 | 87.8832 | 75.5464 | 6.4103 | 6.0762 |
| dairy | international food | 0.1450 | 15.4814 | 12.9998 | 131.4885 | 88.1089 | 64.4671 | 6.4261 | 5.7740 |
| events | bread | 0.1059 | 8.3681 | 4.8108 | 99.3721 | 158.3910 | 128.8670 | 9.9263 | 7.8300 |
| events | canned & pickled | 1.1206 | 20.0165 | 19.9476 | 134.6140 | 156.9432 | 34.2207 | 9.8891 | 4.4653 |
| events | fruit & vegetables | 0.7191 | 17.2525 | 16.0033 | 79.5281 | 158.8850 | 34.1331 | 9.9422 | 4.3249 |
| fruit & vegetables | bread | 0.5868 | 12.3644 | 8.6807 | 141.4147 | 33.1809 | 132.1253 | 4.2681 | 7.9163 |
| fruit & vegetables | canned & pickled | 0.0751 | 7.6313 | 5.2695 | 58.3940 | 35.2655 | 36.3406 | 4.3636 | 4.5938 |
| fruit & vegetables | dairy | 0.3613 | 10.8053 | 7.0845 | 100.6907 | 35.4211 | 89.0897 | 4.3693 | 6.4505 |
| fruit & vegetables | events | 2.6268 | 26.9012 | 35.3657 | 253.7048 | 34.1331 | 158.8850 | 4.3249 | 9.9422 |
| fruit & vegetables | gifts | 4.1216 | 31.2020 | 40.0139 | 235.7818 | 32.1221 | 3595.5916 | 4.2237 | 55.1795 |
| fruit & vegetables | home cleaning | 0.2820 | 11.1499 | 7.2162 | 112.2010 | 35.1904 | 75.5778 | 4.3581 | 6.0774 |
| fruit & vegetables | home outdoor | 1.7650 | 22.9271 | 26.6697 | 243.9330 | 34.0048 | 166.4065 | 4.3028 | 9.7366 |
| fruit & vegetables | international food | 0.1753 | 10.5574 | 6.2214 | 123.0611 | 34.9631 | 64.4866 | 4.3482 | 5.7749 |
| gifts | bread | 5.8374 | 29.6087 | 33.1800 | 62.1844 | 3589.9013 | 134.1996 | 55.1349 | 8.0206 |
| gifts | canned & pickled | 19.3257 | 33.4576 | 34.8906 | 21.6889 | 3716.3020 | 35.2596 | 56.6101 | 4.5125 |
| gifts | fruit & vegetables | 16.5496 | 33.6290 | 34.3409 | 7.3027 | 3595.5916 | 32.1221 | 55.1795 | 4.2237 |
| gifts | home outdoor | 4.2716 | 17.4545 | 16.8812 | 204.3875 | 3165.0375 | 191.9873 | 51.9084 | 10.4563 |
| home cleaning | bread | 0.0455 | 5.6154 | 1.3333 | 82.6786 | 75.6465 | 130.9721 | 6.0813 | 7.8811 |
| home cleaning | canned & pickled | 0.3607 | 10.8696 | 7.6347 | 102.5252 | 75.0585 | 36.0978 | 6.0571 | 4.5789 |
| home cleaning | dairy | 0.0165 | 6.7582 | 3.2583 | 75.2279 | 75.5464 | 87.8832 | 6.0762 | 6.4103 |

Each domain and domain-pair ratings' dispersion statistics for the supermarket dataset contd.

| source | target | total KL-divergenc | mean user KL-divergenc | median user KL-divergenc | variance user KL-divergenc | source Kurtosis rating | target Kurtosis rating | source skewness rating | target skewness rating |
|---|---|---|---|---|---|---|---|---|---|
| home cleaning | fruit & vegetables | 0.1981 | 9.3715 | 6.8456 | 62.3803 | 75.5778 | 35.1904 | 6.0774 | 4.3581 |
| home cleaning | international food | 0.0521 | 7.6893 | 2.8066 | 111.0918 | 75.2303 | 63.9973 | 6.0657 | 5.7524 |
| home outdoor | bread | 0.0461 | 7.5142 | 3.6506 | 97.6236 | 166.3014 | 134.3117 | 9.7343 | 8.0474 |
| home outdoor | canned & pickled | 0.8344 | 18.8026 | 17.7775 | 129.5316 | 165.3755 | 35.5754 | 9.7181 | 4.5391 |
| home outdoor | fruit & vegetables | 0.5194 | 16.1495 | 14.7704 | 82.4525 | 166.4065 | 34.0048 | 9.7366 | 4.3028 |
| home outdoor | gifts | 0.1776 | 2.6581 | 0.0000 | 85.2015 | 191.9873 | 3165.0375 | 10.4563 | 51.9084 |
| international food | bread | 0.1111 | 8.5279 | 2.6630 | 129.1309 | 64.4583 | 132.2087 | 5.7741 | 7.9308 |
| international food | canned & pickled | 0.2045 | 13.5800 | 10.2547 | 128.4001 | 64.5412 | 35.3256 | 5.7773 | 4.5306 |
| international food | dairy | 0.0641 | 9.9652 | 5.2828 | 115.3590 | 64.4671 | 88.1089 | 5.7740 | 6.4261 |
| international food | fruit & vegetables | 0.0850 | 12.0788 | 9.5737 | 85.3462 | 64.4866 | 34.9631 | 5.7749 | 4.3482 |
| international food | home cleaning | 0.0365 | 9.6382 | 4.6167 | 124.7317 | 63.9973 | 75.2303 | 5.7524 | 6.0657 |

## A.3 ERROR OF ALGORITHMS ON DOMAIN PAIRS IN SUPERMARKET DATASET

Table 40: RMSE and MAE for domain-pairs in the supermarket dataset

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| bread | canned & pickled | 0.3079 | 0.5052 | 0.4909 | 0.2903 | 0.5427 | 0.2681 | 0.4161 | 0.4016 | 0.1690 | 0.4414 |
| bread | dairy | 0.3242 | 0.4956 | 0.4934 | 0.2443 | 0.7641 | 0.2857 | 0.4048 | 0.4042 | 0.1470 | 0.3233 |
| bread | events | 0.3251 | 0.4177 | 0.2406 | 0.5641 | 0.4975 | 0.2464 | 0.2738 | 0.1474 | 0.3631 | 0.3774 |
| bread | fruit & vegetables | 0.2909 | 0.4515 | 0.4533 | 0.1527 | 0.3495 | 0.2583 | 0.3637 | 0.3657 | 0.0980 | 0.1678 |
| bread | gifts | 0.3583 | 0.4230 | 0.2784 | 0.6110 | 2.1877 | 0.2742 | 0.2988 | 0.1930 | 0.4172 | 0.4741 |
| bread | home cleaning | 0.2877 | 0.4110 | 0.3692 | 0.2364 | 0.5288 | 0.2437 | 0.3332 | 0.2975 | 0.1331 | 0.4400 |
| bread | home outdoor | 0.3577 | 0.5301 | 0.3505 | 0.5749 | 0.5180 | 0.2700 | 0.4037 | 0.2388 | 0.3740 | 0.3904 |
| bread | international food | 0.3590 | 0.5177 | 0.4491 | 0.3617 | 0.8108 | 0.2977 | 0.4208 | 0.3485 | 0.2121 | 0.4500 |
| canned & pickled | bread | 0.3821 | 0.4889 | 0.4967 | 0.2785 | 0.4732 | 0.3472 | 0.4011 | 0.4102 | 0.1702 | 0.3594 |
| canned & pickled | dairy | 0.3459 | 0.5666 | 0.5790 | 0.2474 | 0.6084 | 0.3111 | 0.4782 | 0.4919 | 0.1514 | 0.2604 |
| canned & pickled | events | 0.3613 | 0.3365 | 0.2567 | 0.5528 | 0.6140 | 0.2828 | 0.2026 | 0.1543 | 0.3436 | 0.3588 |
| canned & pickled | fruit & vegetables | 0.3198 | 0.4607 | 0.4712 | 0.1529 | 0.2677 | 0.2883 | 0.3722 | 0.3824 | 0.0995 | 0.1524 |
| canned & pickled | gifts | 0.3563 | 0.3468 | 0.2633 | 0.5902 | 0.8015 | 0.2858 | 0.2292 | 0.1788 | 0.3915 | 0.4230 |
| canned & pickled | home cleaning | 0.2982 | 0.3896 | 0.3725 | 0.2468 | 0.9035 | 0.2529 | 0.3143 | 0.3005 | 0.1379 | 0.3245 |
| canned & pickled | home outdoor | 0.3688 | 0.3925 | 0.3049 | 0.5768 | 0.5440 | 0.2817 | 0.2767 | 0.2214 | 0.3739 | 0.3695 |
| canned & pickled | international food | 0.3955 | 0.4547 | 0.3717 | 0.3577 | 0.6034 | 0.3174 | 0.3637 | 0.2948 | 0.2054 | 0.4759 |
| dairy | bread | 0.3831 | 0.4711 | 0.4801 | 0.2826 | 0.8269 | 0.3485 | 0.3844 | 0.3930 | 0.1768 | 0.3469 |
| dairy | canned & pickled | 0.3387 | 0.5532 | 0.5319 | 0.3051 | 0.5480 | 0.3001 | 0.4637 | 0.4397 | 0.1827 | 0.4469 |
| dairy | fruit & vegetables | 0.3298 | 0.4473 | 0.4592 | 0.1513 | 0.2886 | 0.3026 | 0.3593 | 0.3710 | 0.0975 | 0.1625 |
| dairy | home cleaning | 0.3075 | 0.5071 | 0.4746 | 0.2407 | 0.5545 | 0.2637 | 0.4216 | 0.3877 | 0.1395 | 0.4520 |
| dairy | international food | 0.3945 | 0.4758 | 0.3795 | 0.3747 | 0.6692 | 0.3177 | 0.3837 | 0.3014 | 0.2247 | 0.4404 |
| events | bread | 0.3791 | 0.3930 | 0.3932 | 0.2671 | 0.4269 | 0.3411 | 0.3165 | 0.3134 | 0.1712 | 0.3058 |
| events | canned & pickled | 0.3417 | 0.3964 | 0.4025 | 0.2641 | 0.4680 | 0.2942 | 0.3207 | 0.3240 | 0.1562 | 0.3575 |
| events | fruit & vegetables | 0.3248 | 0.3549 | 0.3524 | 0.1486 | 0.3930 | 0.2850 | 0.2777 | 0.2737 | 0.0963 | 0.2298 |
| fruit & vegetables | bread | 0.3830 | 0.4779 | 0.4717 | 0.2878 | 0.5003 | 0.3480 | 0.3901 | 0.3843 | 0.1749 | 0.3482 |
| fruit & vegetables | canned & pickled | 0.3424 | 0.5180 | 0.4917 | 0.3047 | 0.5646 | 0.3036 | 0.4291 | 0.4024 | 0.1801 | 0.4361 |
| fruit & vegetables | dairy | 0.3472 | 0.4979 | 0.4917 | 0.2355 | 0.5321 | 0.3144 | 0.4065 | 0.4014 | 0.1417 | 0.4053 |
| fruit & vegetables | events | 0.3528 | 0.6251 | 0.2484 | 0.5661 | 0.4967 | 0.2772 | 0.5294 | 0.1487 | 0.3700 | 0.3904 |
| fruit & vegetables | gifts | 0.3687 | 0.5832 | 0.2843 | 0.5981 | 0.6281 | 0.2906 | 0.4949 | 0.1937 | 0.4070 | 0.3735 |
| fruit & vegetables | home cleaning | 0.2993 | 0.5435 | 0.4736 | 0.2502 | 0.5393 | 0.2544 | 0.4597 | 0.3870 | 0.1416 | 0.4422 |
| fruit & vegetables | home outdoor | 0.3693 | 0.5561 | 0.3270 | 0.5913 | 0.6861 | 0.2821 | 0.4722 | 0.2315 | 0.4021 | 0.3479 |
| fruit & vegetables | international food | 0.3903 | 0.4932 | 0.3822 | 0.3938 | 0.5942 | 0.3136 | 0.4081 | 0.3060 | 0.2371 | 0.4953 |
| gifts | bread | 0.3665 | 0.3938 | 0.4052 | 0.2617 | 0.4166 | 0.3273 | 0.3159 | 0.3260 | 0.1631 | 0.3037 |
| gifts | canned & pickled | 0.3226 | 0.3968 | 0.3995 | 0.2691 | 0.4597 | 0.2796 | 0.3201 | 0.3209 | 0.1637 | 0.3556 |
| gifts | fruit & vegetables | 0.3027 | 0.3468 | 0.3607 | 0.1481 | 0.6015 | 0.2656 | 0.2712 | 0.2809 | 0.0956 | 0.1791 |
| gifts | home outdoor | 0.3465 | 0.3402 | 0.3339 | 0.5783 | 1.1719 | 0.2627 | 0.2295 | 0.2241 | 0.3753 | 0.4182 |

RMSE and MAE for domain-pairs in the supermarket dataset contd.

| source | target | CCA RMSE | CD-SVD RMSE | SD-SVD RMSE | RMGM RMSE | CMF RMSE | CCA MAE | CD-SVD MAE | SD-SVD MAE | RMGM MAE | CMF MAE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| home cleaning | bread | 0.3852 | 0.3775 | 0.4031 | 0.2858 | 0.4801 | 0.3504 | 0.3002 | 0.3228 | 0.1739 | 0.3649 |
| home cleaning | canned & pickled | 0.3466 | 0.4024 | 0.3971 | 0.2948 | 0.5565 | 0.3068 | 0.3246 | 0.3212 | 0.1711 | 0.4448 |
| home cleaning | dairy | 0.3547 | 0.4922 | 0.4913 | 0.2382 | 0.5311 | 0.3222 | 0.4024 | 0.4019 | 0.1448 | 0.4049 |
| home cleaning | fruit & vegetables | 0.3347 | 0.5020 | 0.5170 | 0.1502 | 0.2700 | 0.3076 | 0.4114 | 0.4274 | 0.0971 | 0.1549 |
| home cleaning | international food | 0.4043 | 0.5097 | 0.4548 | 0.3700 | 0.6406 | 0.3224 | 0.4068 | 0.3518 | 0.2151 | 0.4581 |
| home outdoor | bread | 0.3517 | 0.4847 | 0.4857 | 0.2629 | 0.4540 | 0.3119 | 0.3979 | 0.3985 | 0.1652 | 0.3167 |
| home outdoor | canned & pickled | 0.3212 | 0.4052 | 0.3897 | 0.2620 | 0.4637 | 0.2778 | 0.3279 | 0.3151 | 0.1587 | 0.3594 |
| home outdoor | fruit & vegetables | 0.3034 | 0.3495 | 0.3511 | 0.1473 | 0.3537 | 0.2659 | 0.2713 | 0.2745 | 0.0960 | 0.2188 |
| home outdoor | gifts | 0.3232 | 0.3128 | 0.3057 | 0.6024 | 0.6012 | 0.2438 | 0.1982 | 0.1942 | 0.4018 | 0.3945 |
| international food | bread | 0.3912 | 0.4873 | 0.4867 | 0.2956 | 0.4548 | 0.3549 | 0.3994 | 0.3996 | 0.1782 | 0.3189 |
| international food | canned & pickled | 0.3356 | 0.3873 | 0.3962 | 0.2722 | 0.4583 | 0.2955 | 0.3141 | 0.3202 | 0.1603 | 0.3579 |
| international food | dairy | 0.3484 | 0.3968 | 0.4101 | 0.2261 | 0.4811 | 0.3123 | 0.3173 | 0.3293 | 0.1416 | 0.3597 |
| international food | fruit & vegetables | 0.3219 | 0.3468 | 0.3551 | 0.1462 | 0.3009 | 0.2883 | 0.2685 | 0.2770 | 0.0952 | 0.1604 |
| international food | home cleaning | 0.2971 | 0.4891 | 0.4688 | 0.2414 | 0.4078 | 0.2522 | 0.4012 | 0.3816 | 0.1331 | 0.2649 |

Table 41: Significant RMSE relations of algorithms in each domain pair for the Supermarket dataset

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bread | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 10 |
| bread | dairy | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| bread | events | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 10 |
| bread | fruit & vegetables | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| bread | gifts | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 5 |
| bread | home cleaning | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 10 |
| bread | home outdoor | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 7 |
| bread | international food | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 9 |
| canned & pickled | bread | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| canned & pickled | dairy | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 |
| canned & pickled | events | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 8 |
| canned & pickled | fruit & vegetables | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| canned & pickled | gifts | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 8 |
| canned & pickled | home cleaning | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| canned & pickled | home outdoor | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 8 |
| canned & pickled | international food | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 8 |
| dairy | bread | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| dairy | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 8 |
| dairy | fruit & vegetables | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 10 |
| dairy | home cleaning | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 10 |
| dairy | international food | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 7 |
| events | bread | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| events | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| events | fruit & vegetables | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 6 |
| fruit & vegetables | bread | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| fruit & vegetables | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 10 |
| fruit & vegetables | dairy | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| fruit & vegetables | events | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 10 |
| fruit & vegetables | gifts | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 7 |
| fruit & vegetables | home cleaning | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 9 |
| fruit & vegetables | home outdoor | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 5 |
| fruit & vegetables | international food | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 7 |
| gifts | bread | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| gifts | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| gifts | fruit & vegetables | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| gifts | home outdoor | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 3 |
| home cleaning | bread | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 9 |
| home cleaning | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| home cleaning | dairy | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |

Significant RMSE relations of algorithms in each domain pair for the Supermarket dataset contd.

| source | target | CD-CCA >CD-SVD | CD-SVD >CD-CCA | CD-CCA >SD-SVD | SD-SVD >CD-CCA | CD-CCA >CMF | CMF >CD-CCA | CD-CCA >RMGM | RMGM >CDCCA | CMF >SD-SVD | SD-SVD >CMF | CMF >CD-SVD | CD-SVD >CMF | RMGM >SD-SVD | SD-SVD >RMGM | RMGM >CD-SVD | CD-SVD >RMGM | CD-SVD >SD-SVD | SD-SVD >CD-SVD | CMF >RMGM | RMGM >CMF | sum |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| home cleaning | fruit & vegetables | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 10 |
| home cleaning | international food | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 9 |
| home outdoor | bread | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| home outdoor | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| home outdoor | fruit & vegetables | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| home outdoor | gifts | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 6 |
| international food | bread | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| international food | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| international food | dairy | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 10 |
| international food | fruit & vegetables | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| international food | home cleaning | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |

Table 42: Significant MAE relations of algorithms in each domain pair for the Supermarket dataset

| | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bread | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| bread | dairy | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| bread | events | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 8 |
| bread | fruit & vegetables | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| bread | gifts | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 6 |
| bread | home cleaning | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 10 |
| bread | home outdoor | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 6 |
| bread | international food | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 10 |
| canned & pickled | bread | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| canned & pickled | dairy | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| canned & pickled | events | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 9 |
| canned & pickled | fruit & vegetables | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| canned & pickled | gifts | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 9 |
| canned & pickled | home cleaning | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| canned & pickled | home outdoor | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 8 |
| canned & pickled | international food | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 10 |
| dairy | bread | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| dairy | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 8 |
| dairy | fruit & vegetables | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 10 |
| dairy | home cleaning | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 9 |
| dairy | international food | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 9 |
| events | bread | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| events | canned & pickled | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| events | fruit & vegetables | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| fruit & vegetables | bread | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| fruit & vegetables | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 9 |
| fruit & vegetables | dairy | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| fruit & vegetables | events | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 9 |
| fruit & vegetables | gifts | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 9 |
| fruit & vegetables | home cleaning | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 9 |
| fruit & vegetables | home outdoor | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 10 |
| fruit & vegetables | international food | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 9 |
| gifts | bread | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 5 |
| gifts | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| gifts | fruit & vegetables | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| gifts | home outdoor | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 6 |
| home cleaning | bread | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 9 |
| home cleaning | canned & pickled | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| home cleaning | dairy | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |

Significant MAE relations of algorithms in each domain pair for the Supermarket dataset contd.

| source | target | CD-CCA >CD-SVD | CD-SVD >CD-CCA | CD-CCA >SD-SVD | SD-SVD >CD-CCA | CD-CCA >CMF | CMF >CD-CCA | CD-CCA >RMGM | RMGM >CDCCA | CMF >SD-SVD | SD-SVD >CMF | CMF >CD-SVD | CD-SVD >CMF | RMGM >SD-SVD | SD-SVD >RMGM | RMGM >CD-SVD | CD-SVD >RMGM | CD-SVD >SD-SVD | SD-SVD >CD-SVD | CMF >RMGM | RMGM >CMF | sum |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| home cleaning | fruit & vegetables | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 10 |
| home cleaning | international food | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 9 |
| home outdoor | bread | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| home outdoor | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| home outdoor | fruit & vegetables | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 7 |
| home outdoor | gifts | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 8 |
| international food | bread | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| international food | canned & pickled | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 9 |
| international food | dairy | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| international food | fruit & vegetables | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |
| international food | home cleaning | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 8 |

# A.4 COLD-START ANALYSIS OF DOMAIN PAIRS FOR SUPERMARKET DATASET

## A.4.1 MAEs for Target User Profiles

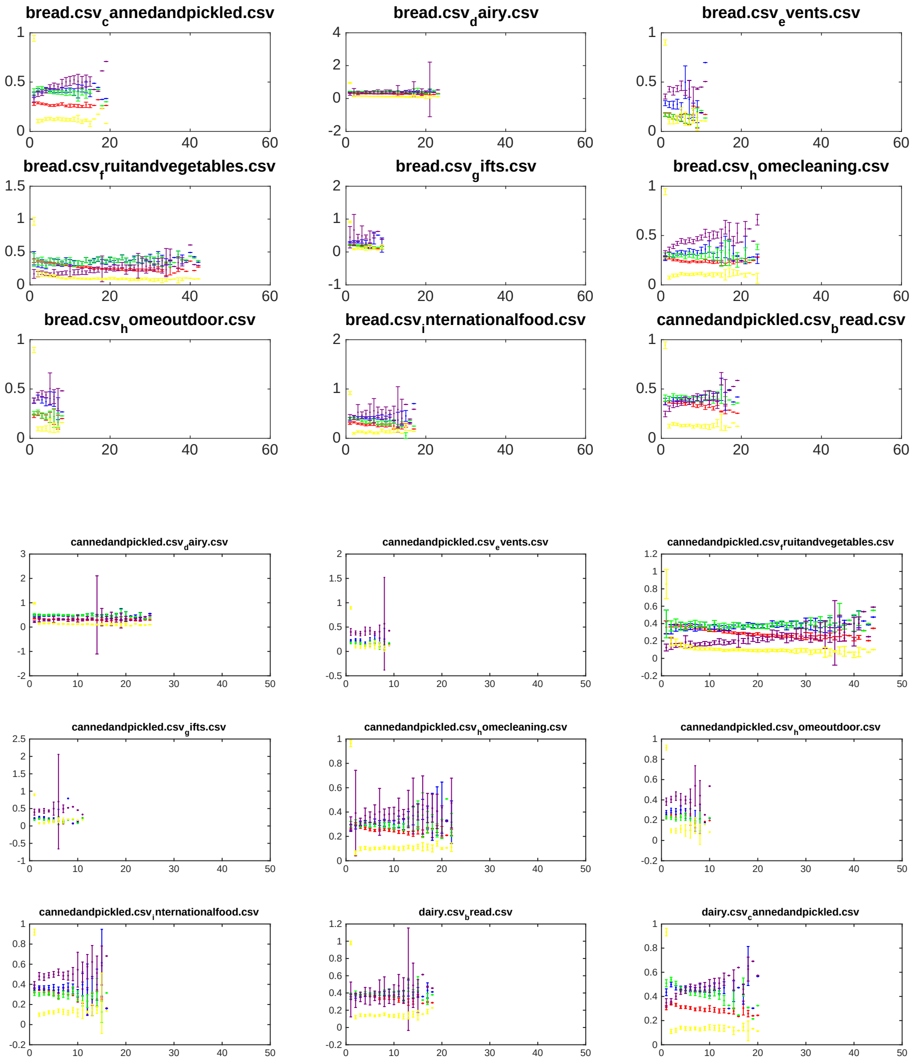


Figure 52: User-based MAE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' target domain profile size

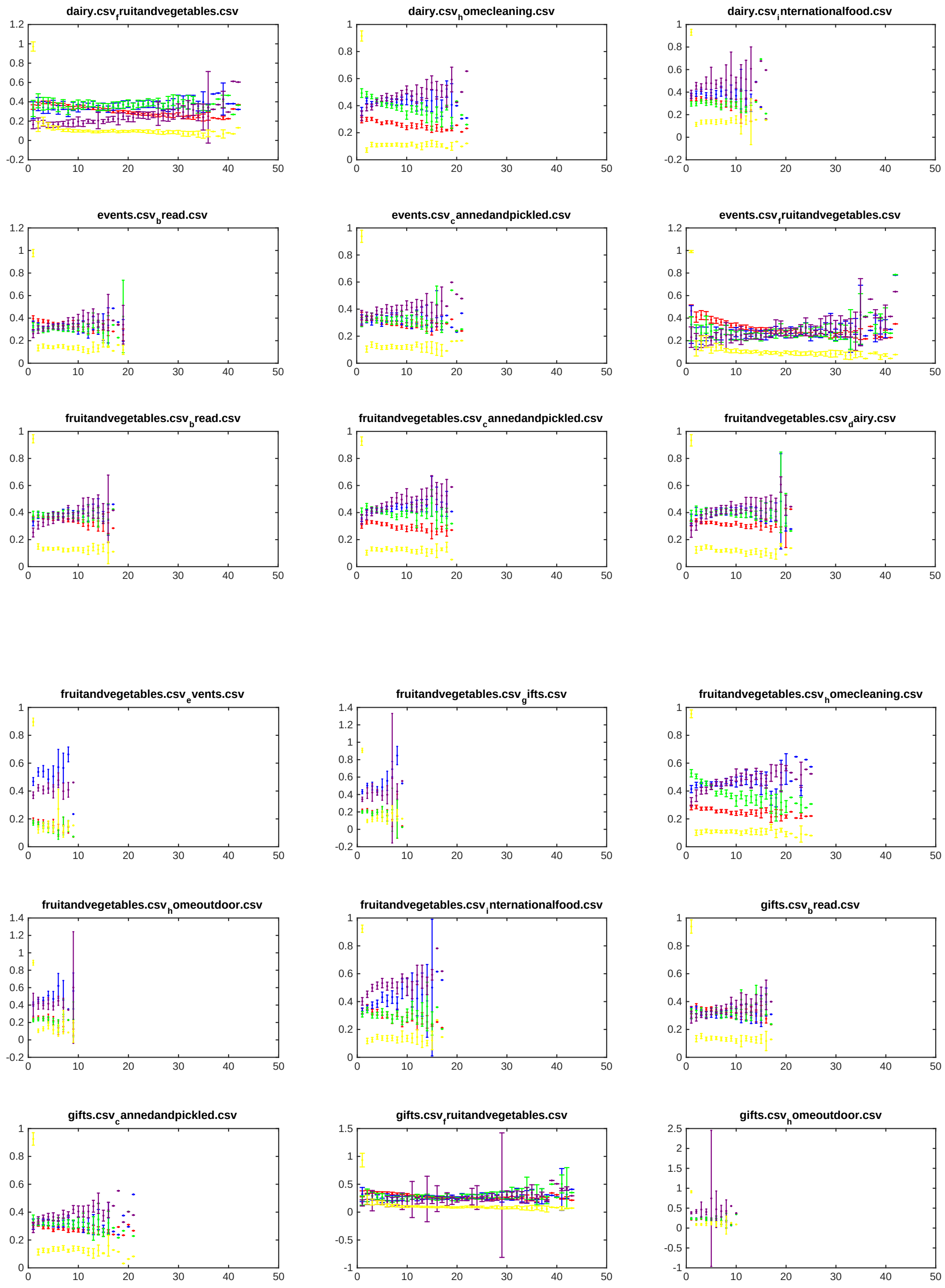


Figure 52: User-based MAE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 19 to 36

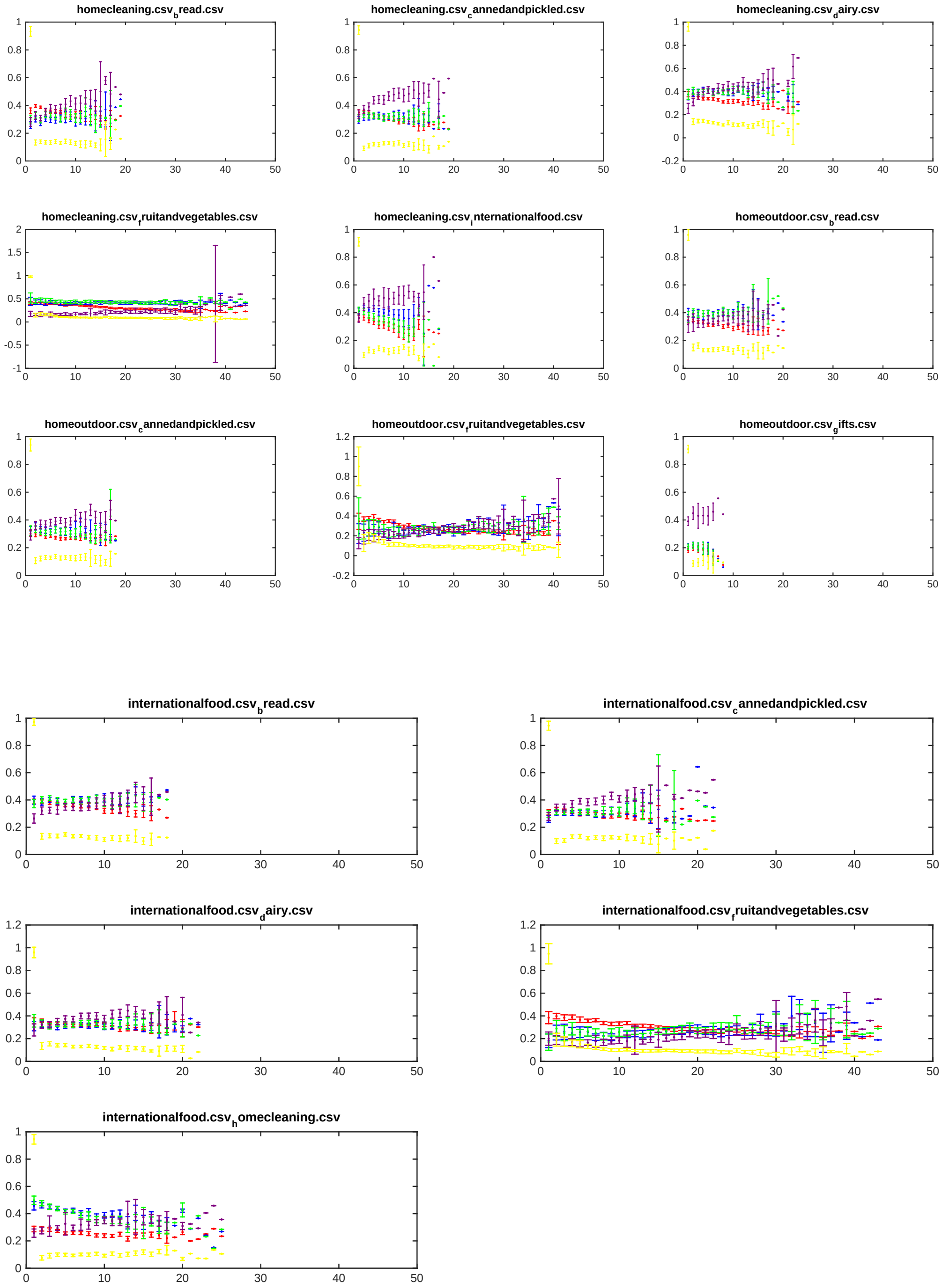


Figure 52: User-based MAE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 37 to 50

### A.4.2 RMSEs for Target User Profiles

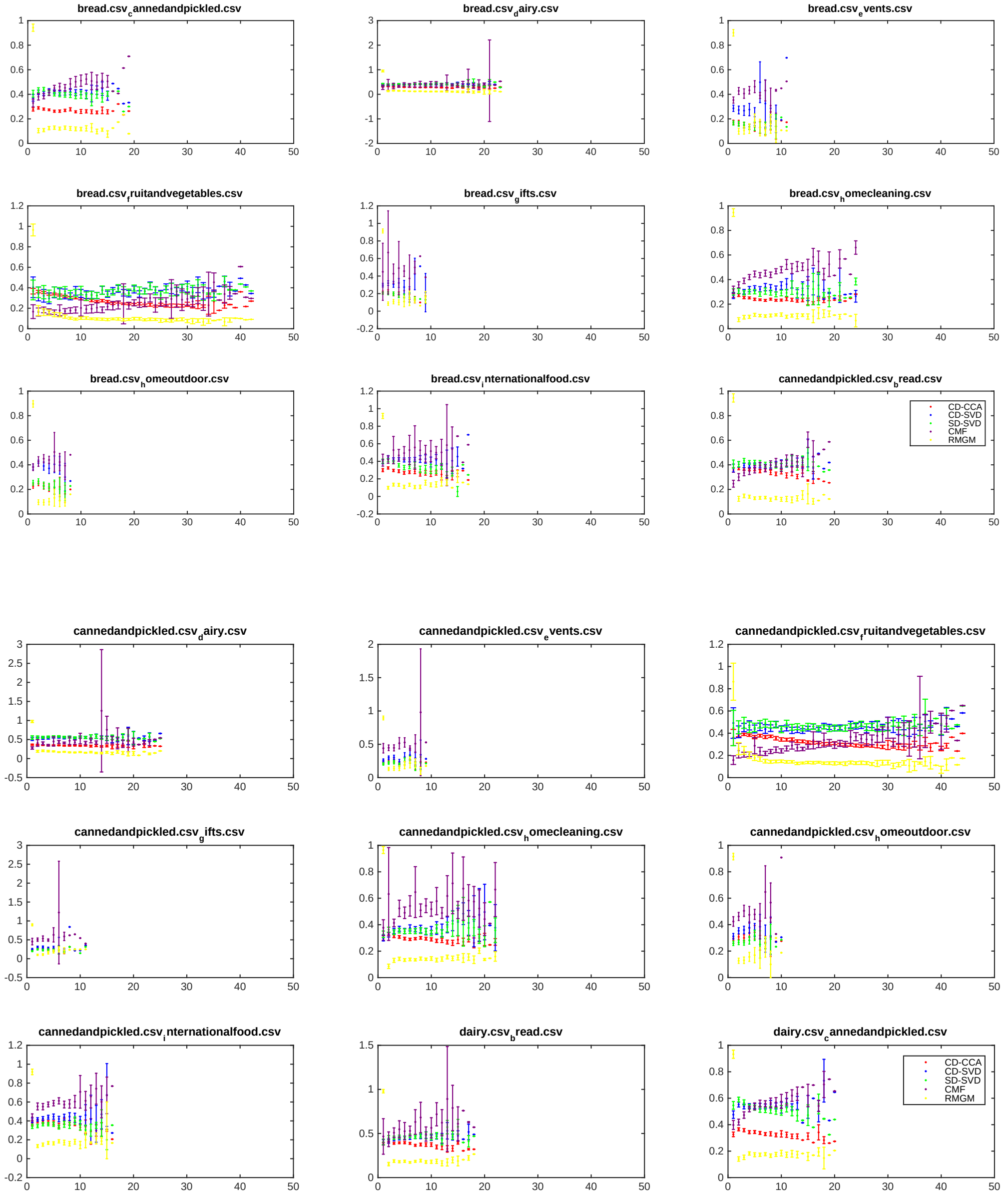


Figure 53: User-based RMSE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Target domain profile size

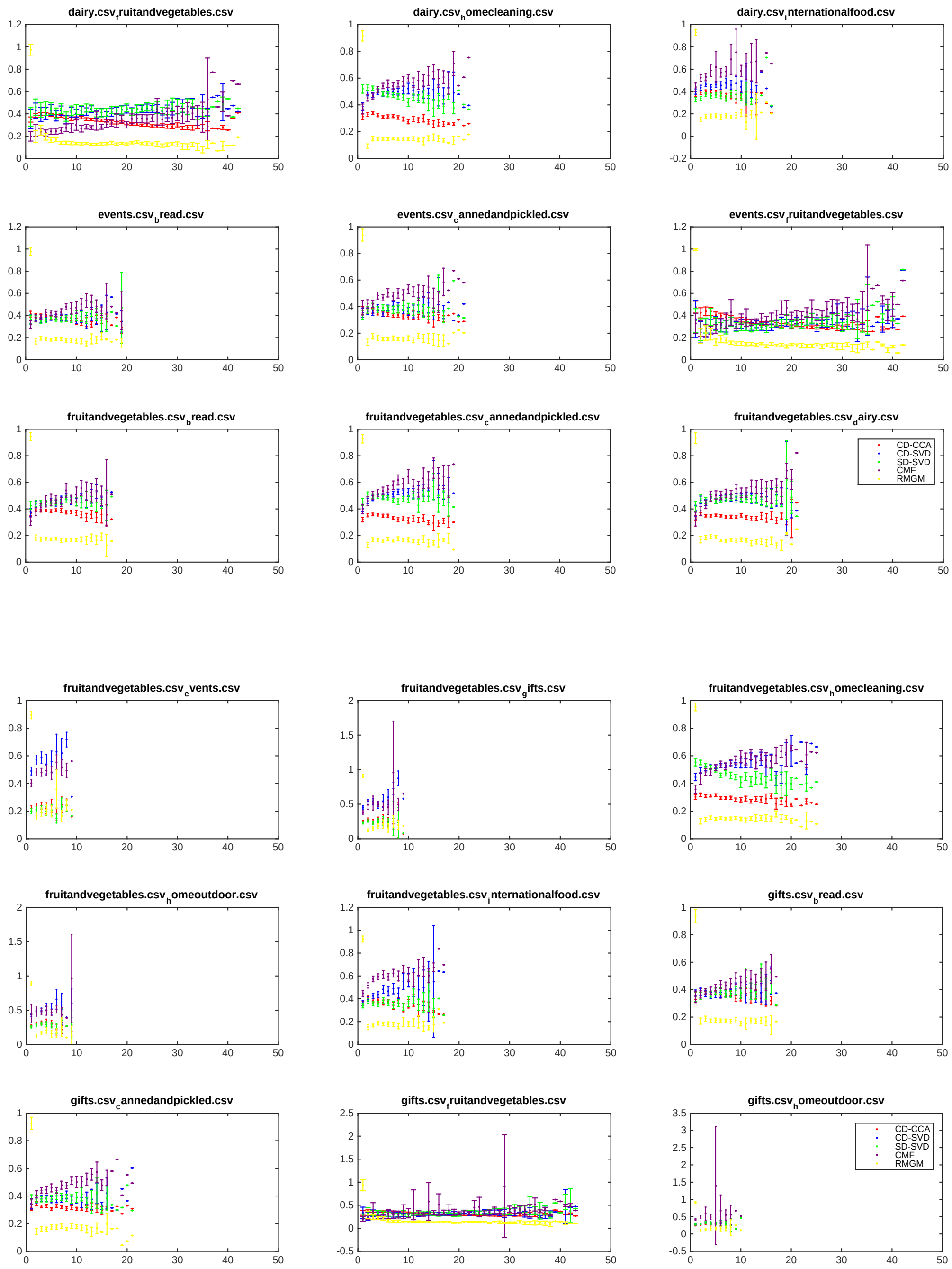
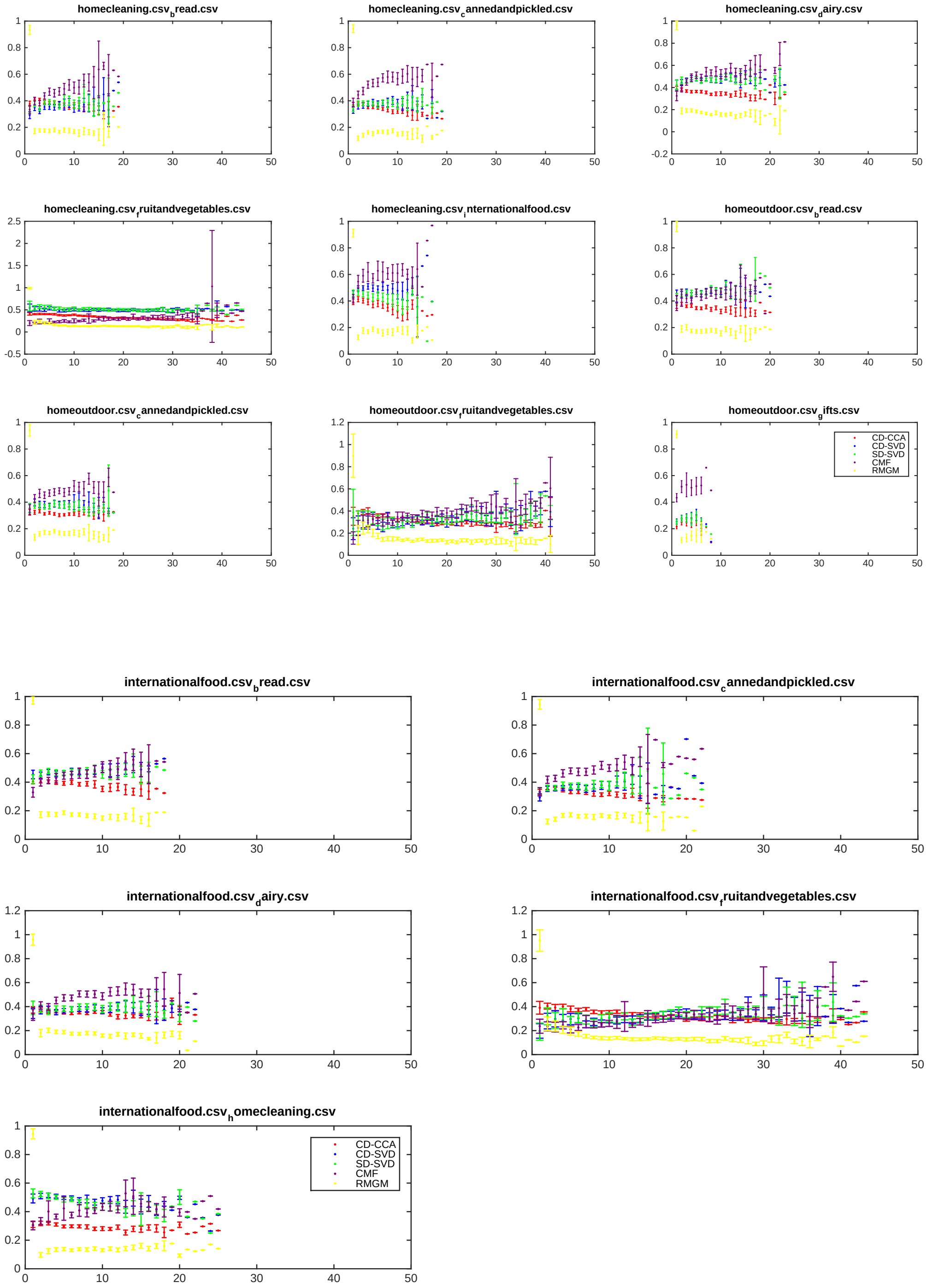


Figure 53: User-based RMSE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Target domain profile size for domain pairs 19 to 36


homecleaning.csv$_{b}$read.csv

homecleaning.csv$_{c}$annedandpickled.csv

homecleaning.csv$_{d}$airy.csv

homecleaning.csv$_{f}$ruitandvegetables.csv

homecleaning.csv$_{i}$nternationalfood.csv

homeoutdoor.csv$_{b}$read.csv

homeoutdoor.csv$_{c}$annedandpickled.csv

homeoutdoor.csv$_{f}$ruitandvegetables.csv

homeoutdoor.csv$_{g}$ifts.csv

internationalfood.csv$_{b}$read.csv

internationalfood.csv$_{c}$annedandpickled.csv

internationalfood.csv$_{d}$airy.csv

internationalfood.csv$_{f}$ruitandvegetables.csv

internationalfood.csv$_{h}$omecleaning.csv

Figure 53: User-based RMSE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Target domain profile size for domain pairs 37 to 50

### A.4.3 RMSEs for Source User Profiles

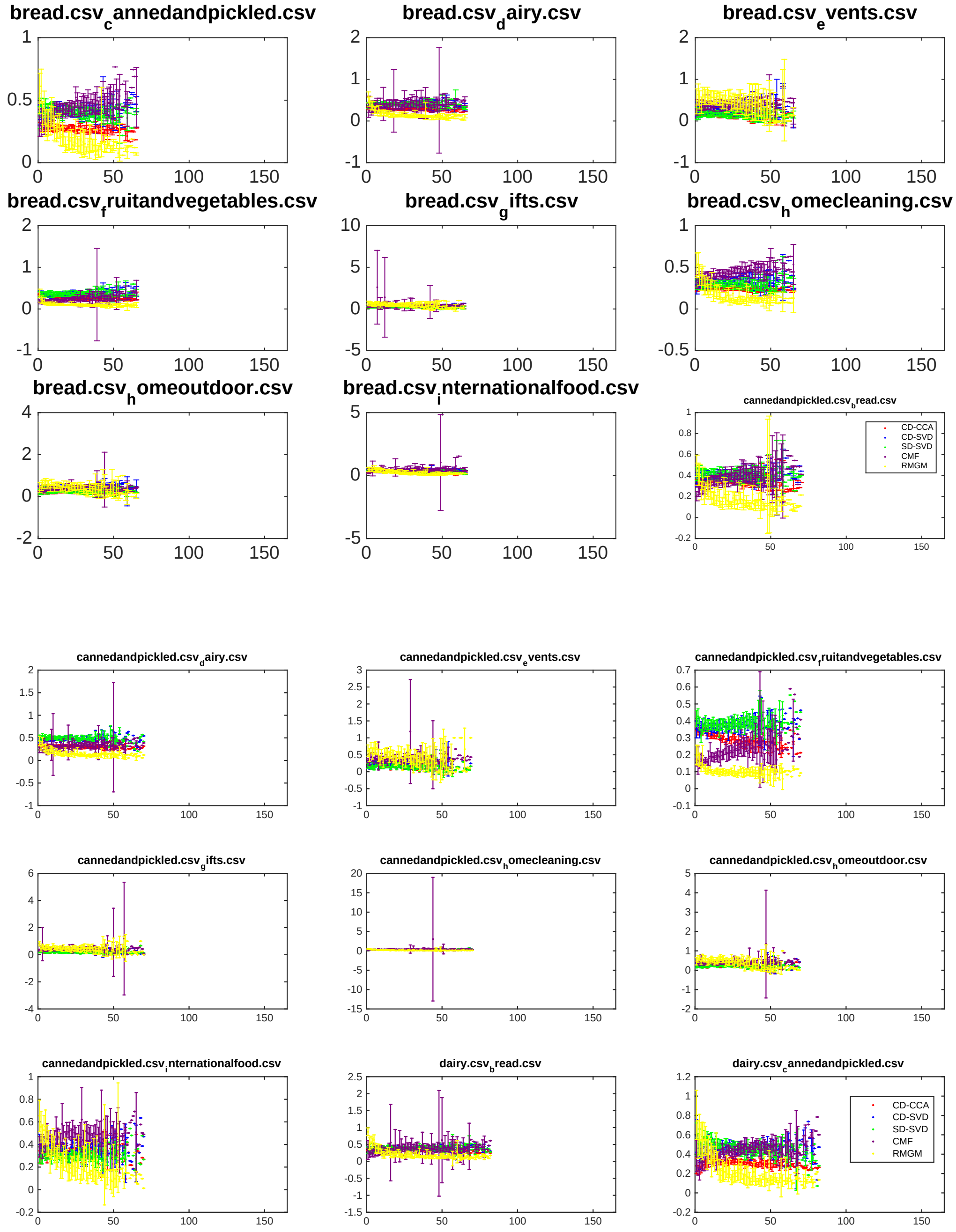


Figure 54: User-based MAE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size

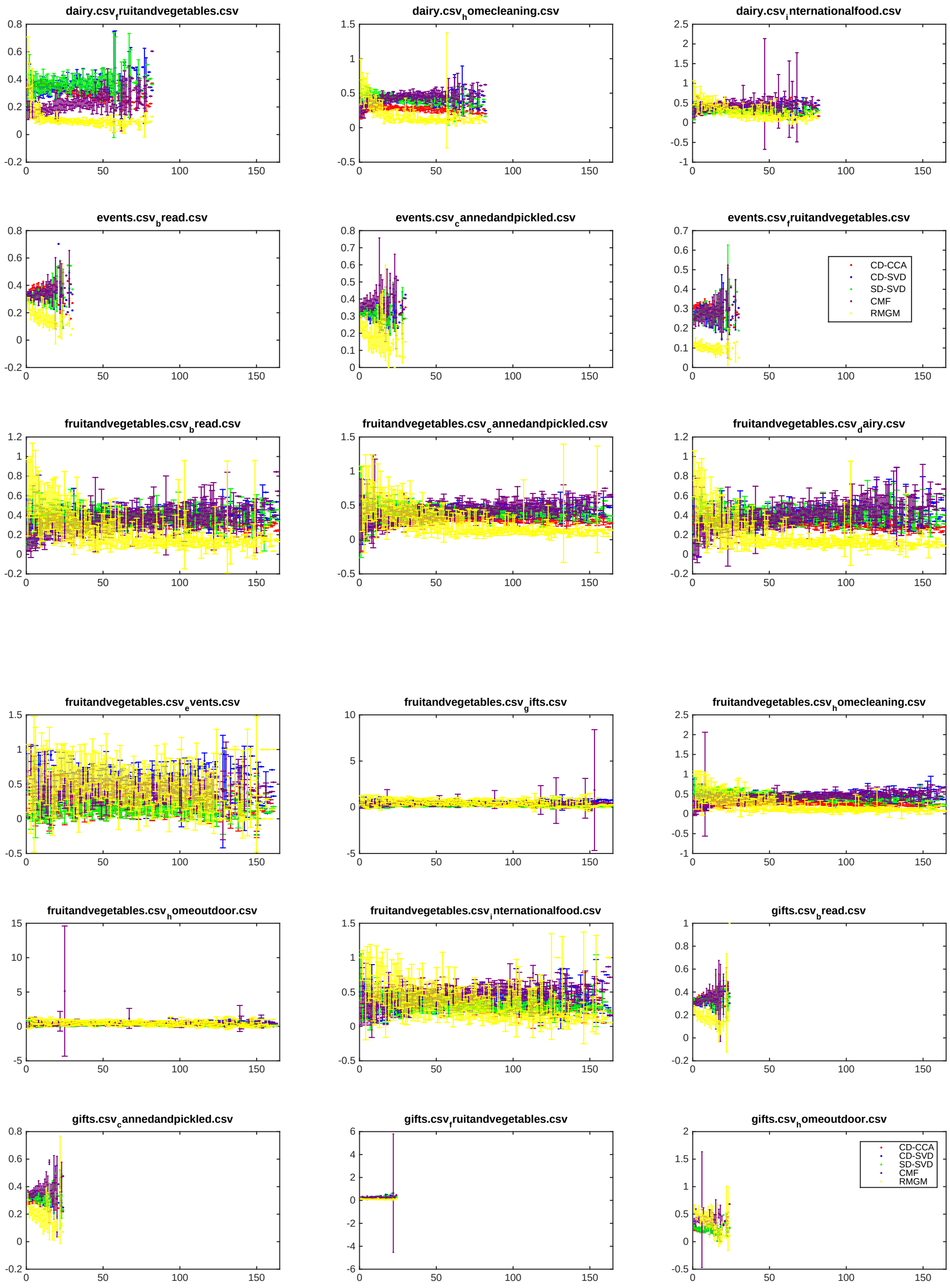


Figure 54: User-based MAE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size for domain pairs 19 to 36

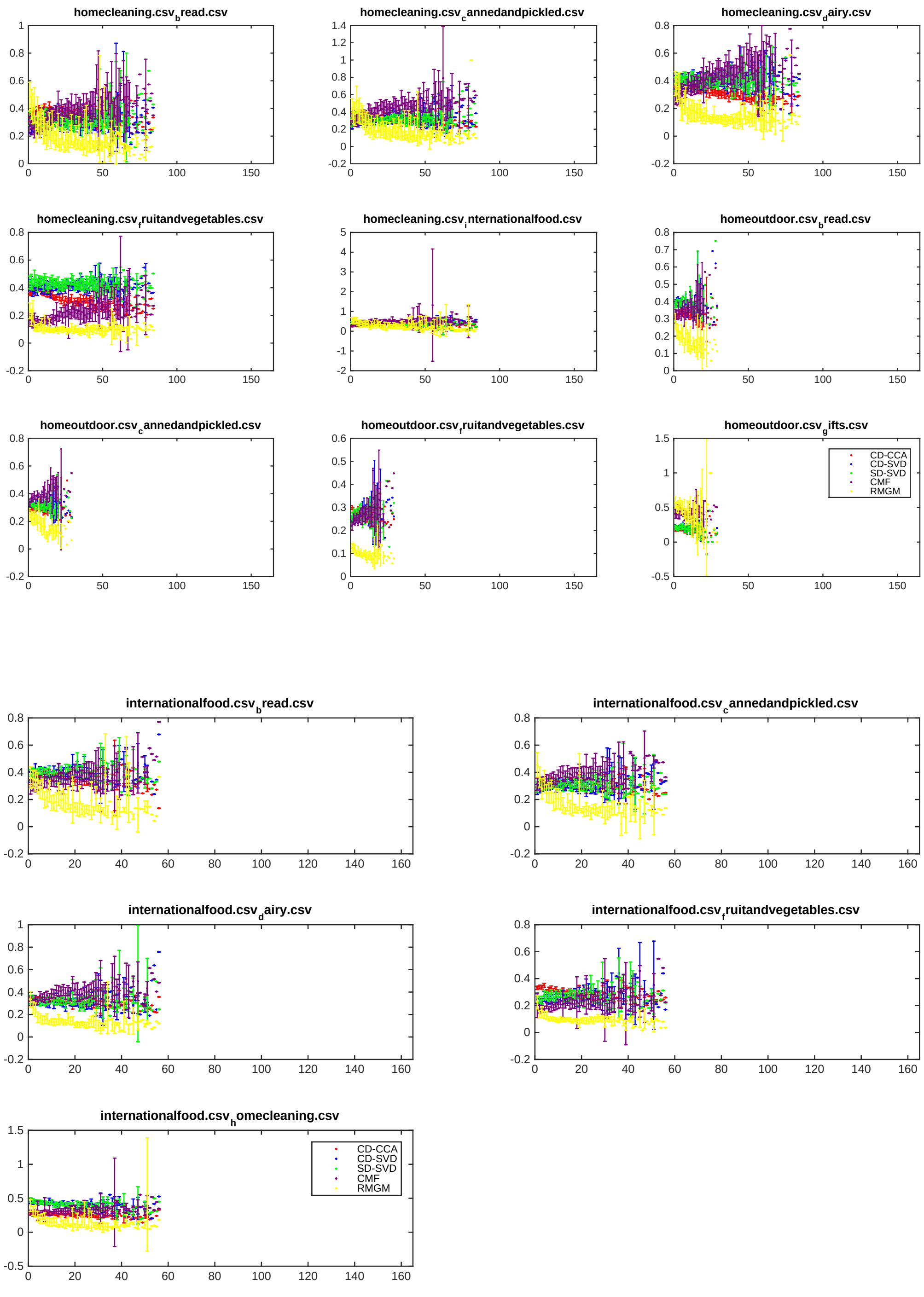


Figure 54: User-based MAE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size for domain pairs 37 to 50

### A.4.4 RMSEs for Source User Profiles

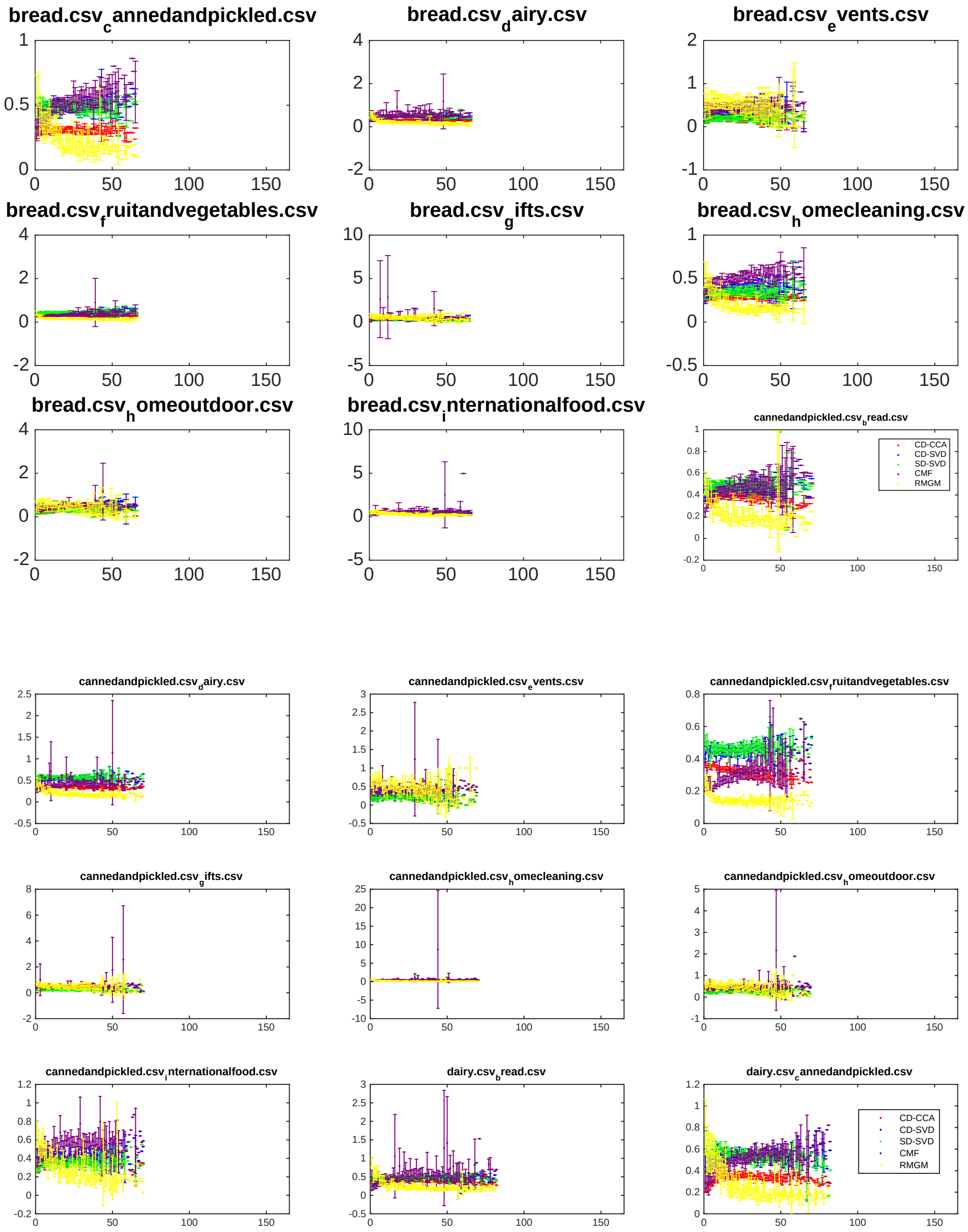


Figure 55: User-based RMSE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size

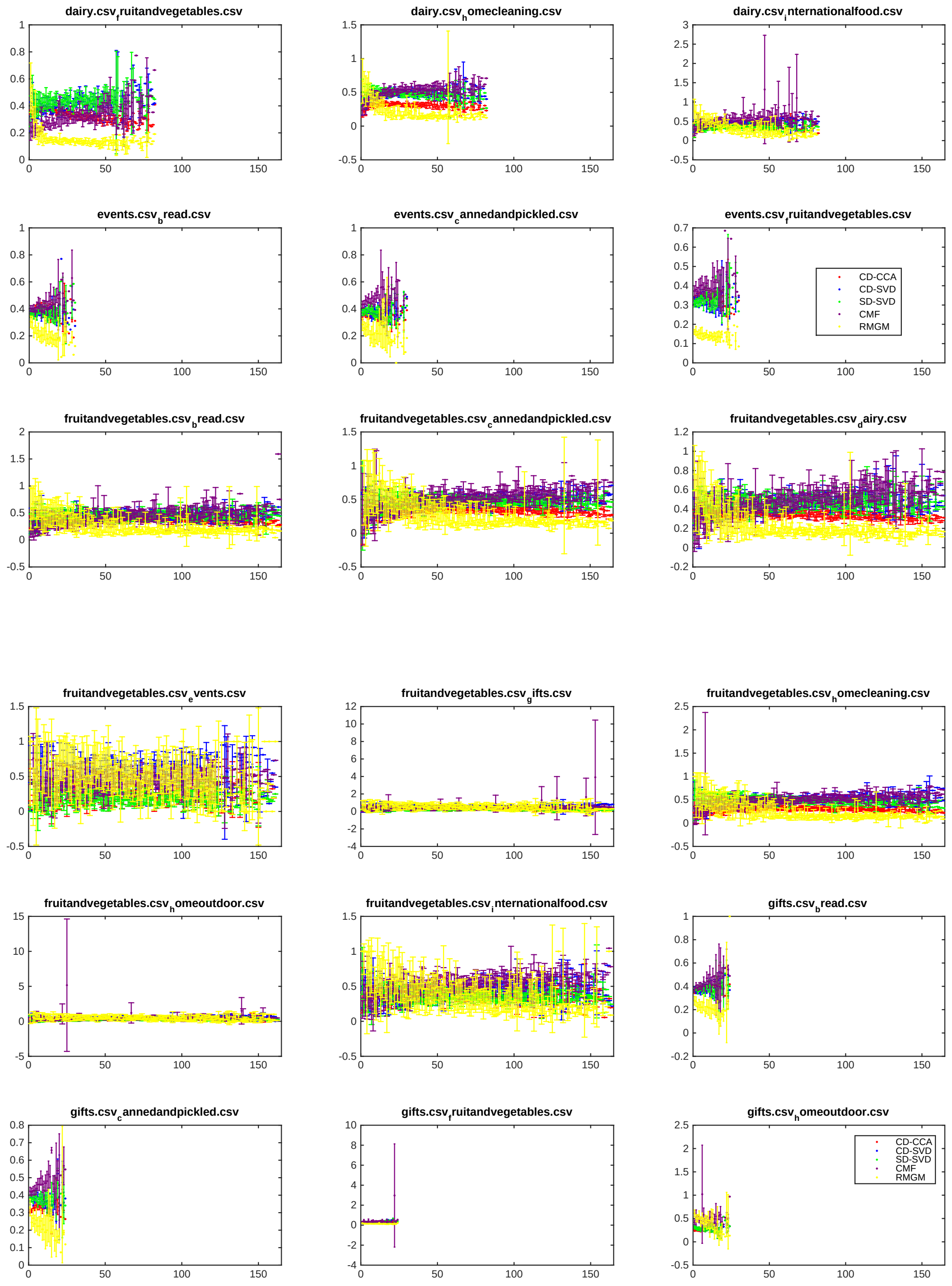


Figure 55: User-based RMSE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size for domain pairs 19 to 36

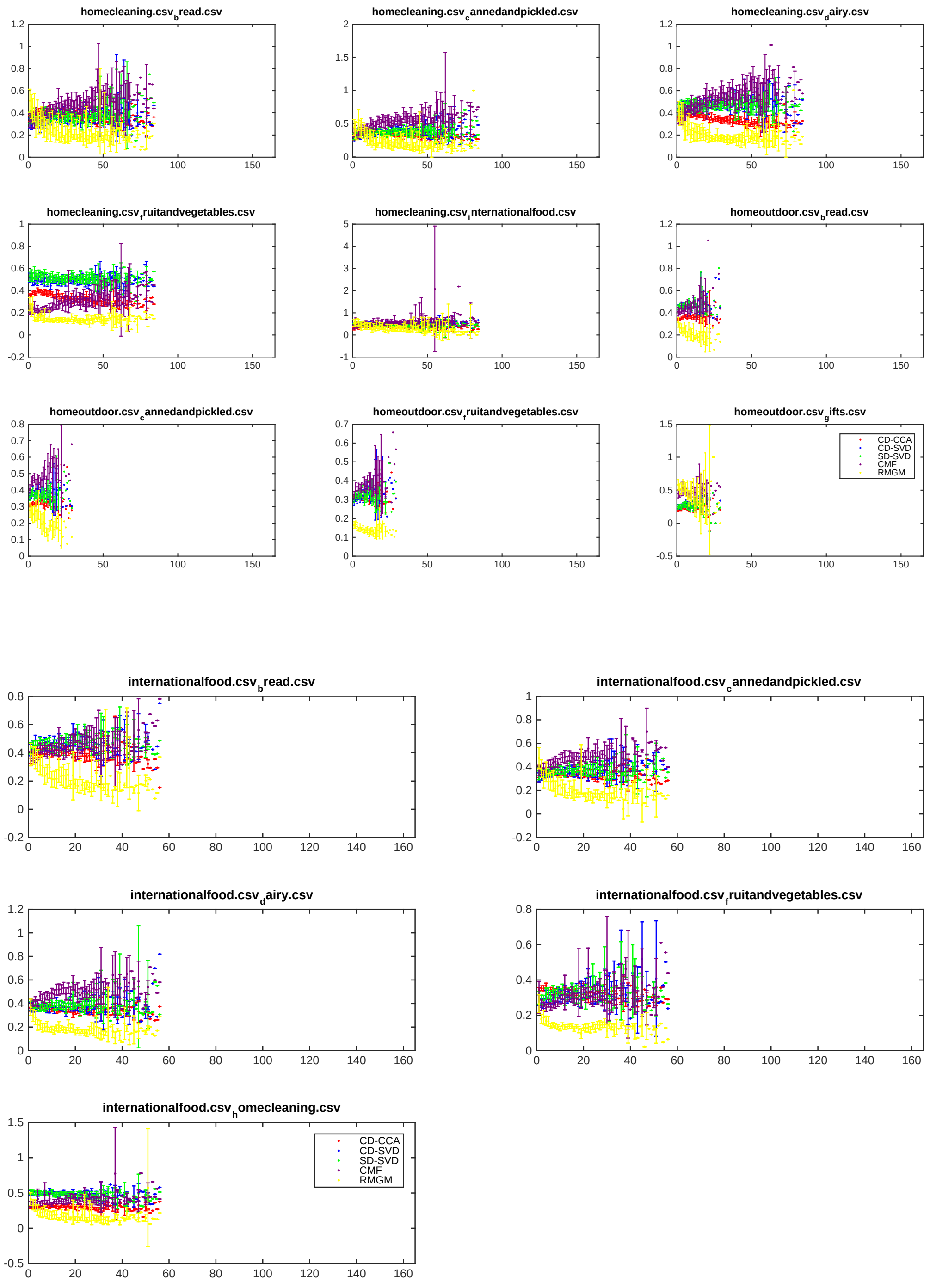


Figure 55: User-based RMSE of algorithms in the supermarket dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size for domain pairs 37 to 50

## A.5 CORRELATION ANALYSIS FOR SUPERMARKET DATASET

### A.5.1 Correlation Analysis Plots for the RMSE of Algorithms in the Supermarket Dataset

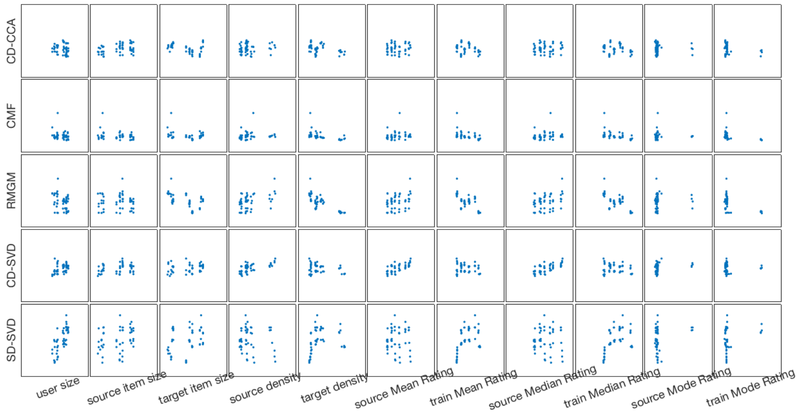


Figure 56: Scatter plot of general and central tendency statistics with the RMSE of algorithms in the Supermarket dataset

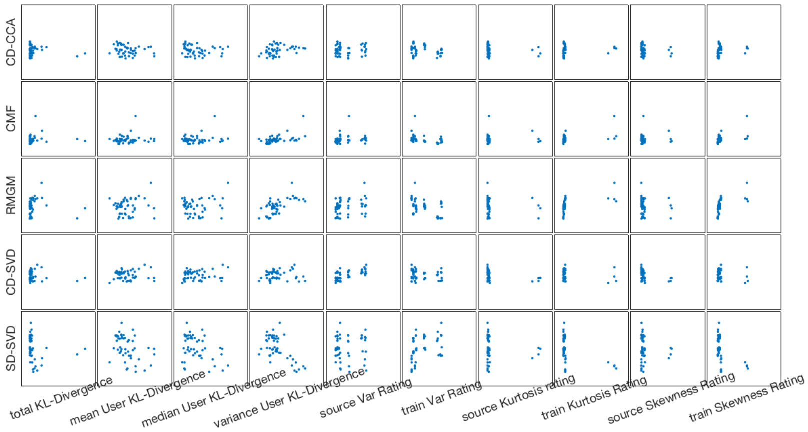


Figure 57: Scatter plot of dispersion statistics with the RMSE of algorithms in the Supermarket dataset

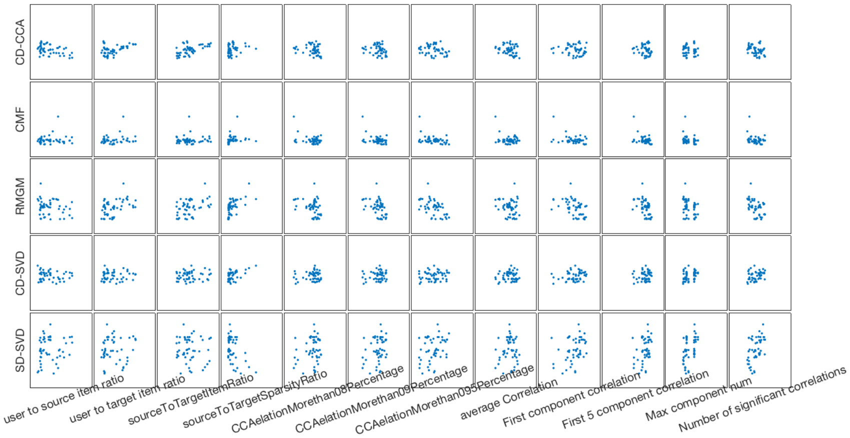


Figure 58: Scatter plot of CCA-related statistics with the RMSE of algorithms in the Supermarket dataset

### A.5.2 Correlation Analysis Plots for the Improvement Ratio of Algorithms in the Supermarket Dataset

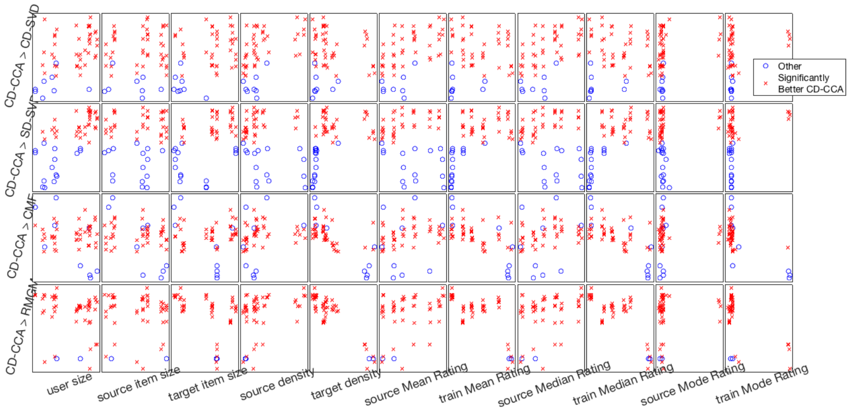


Figure 59: Scatter plot of general and central tendency statistics with the improvement ratio of CD-CCA over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-CCA is significantly better than othor algorithms

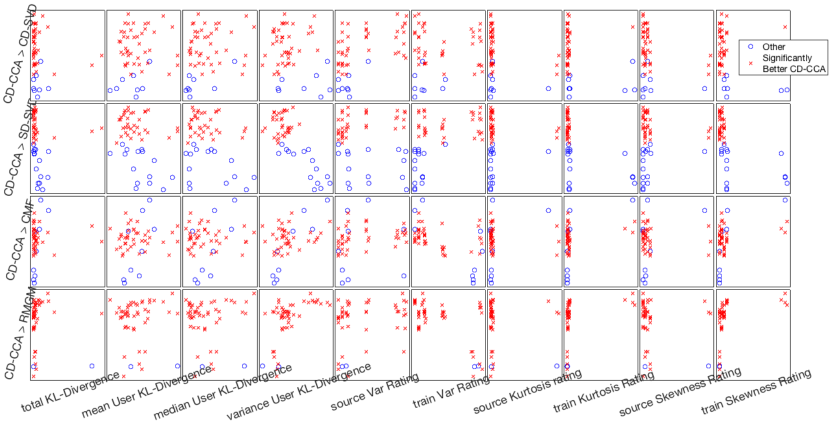


Figure 60: Scatter plot of dispersion statistics with the improvement ratio of CD-CCA over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-CCA is significantly better than othor algorithms

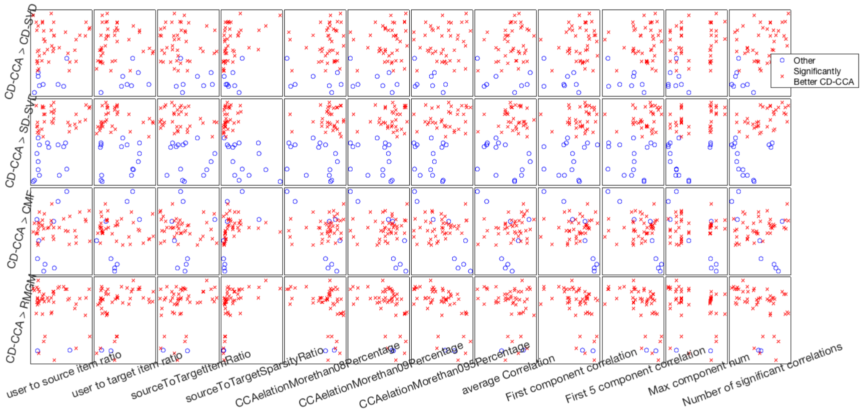


Figure 61: Scatter plot of CCA-related statistics with the improvement ratio of CD-CCA over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-CCA is significantly better than othor algorithms

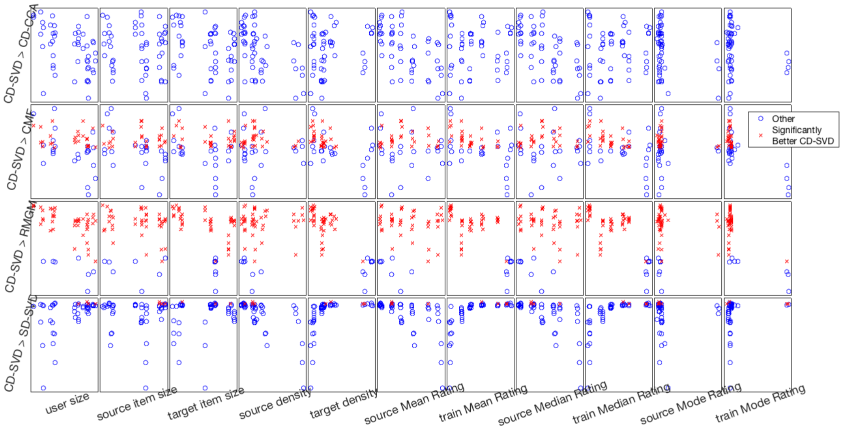


Figure 62: Scatter plot of general and central tendency statistics with the improvement ratio of CD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-SVD is significantly better than othor algorithms

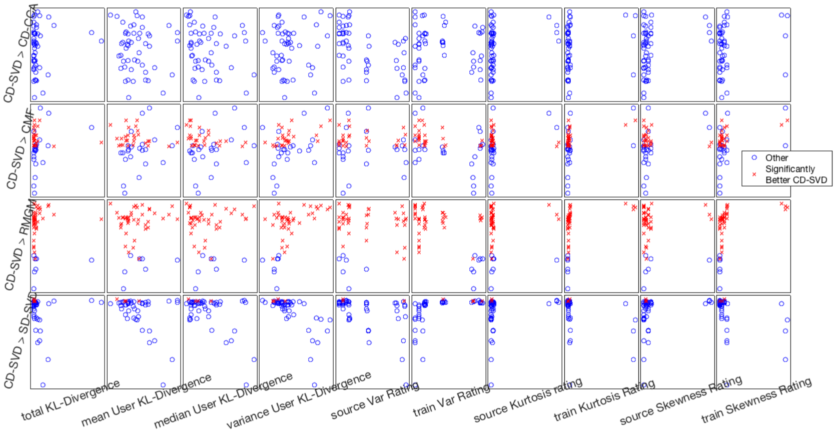


Figure 63: Scatter plot of dispersion statistics with the improvement ratio of CD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-SVD is significantly better than othor algorithms

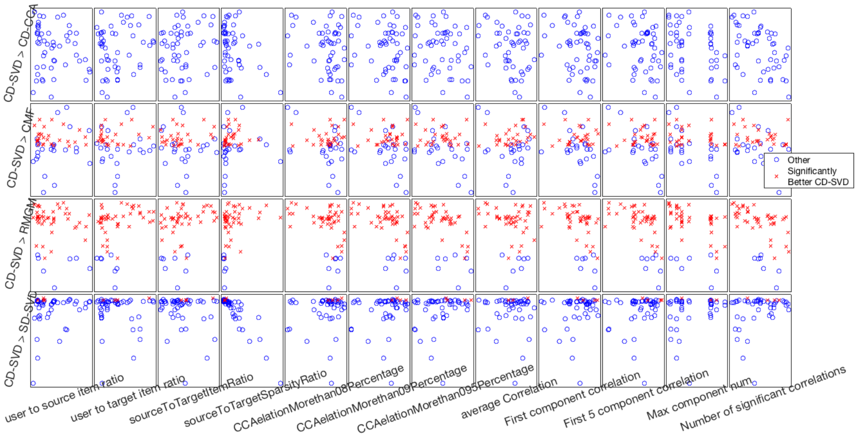


Figure 64: Scatter plot of CCA-related statistics with the improvement ratio of CD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which CD-SVD is significantly better than othor algorithms

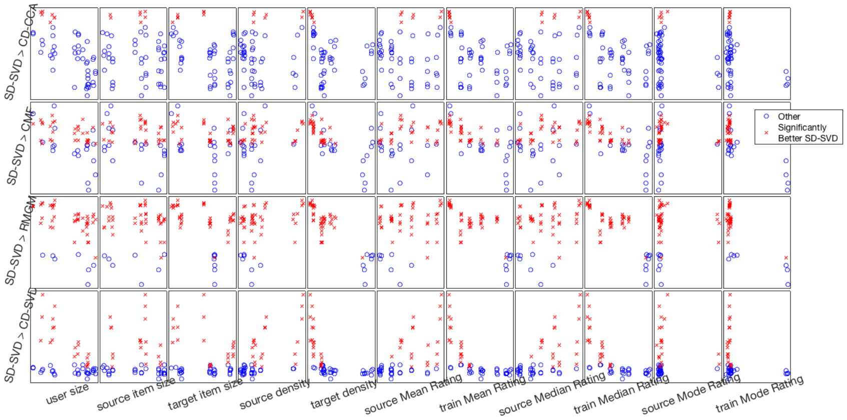


Figure 65: Scatter plot of general and central tendency statistics with the improvement ratio of SD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which SD-SVD is significantly better than othor algorithms

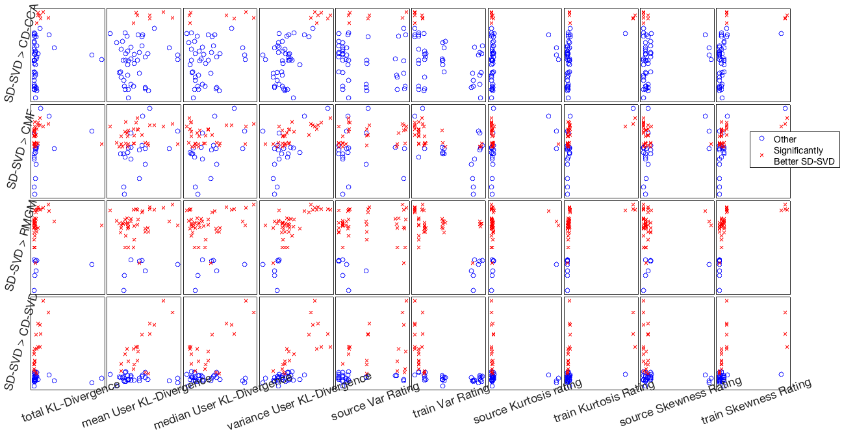


Figure 66: Scatter plot of dispersion statistics with the improvement ratio of SD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which SD-SVD is significantly better than othor algorithms

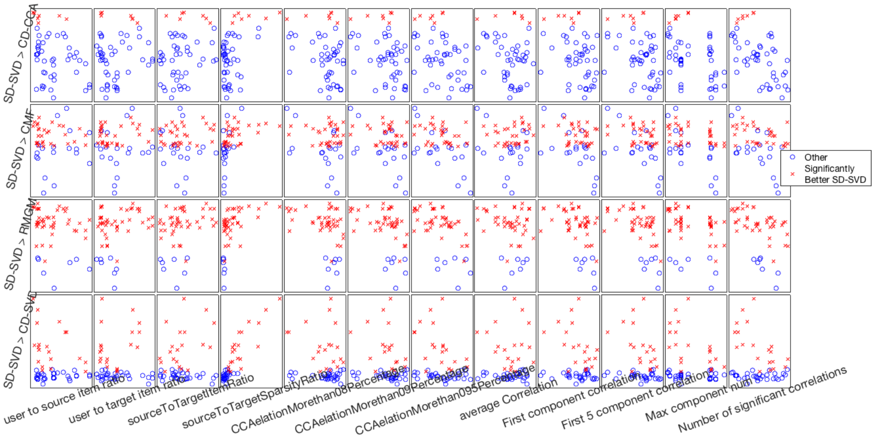


Figure 67: Scatter plot of CCA-related statistics with the improvement ratio of SD-SVD over other algorithms in the Supermarket dataset; red cross shows the cases in which SD-SVD is significantly better than othor algorithms

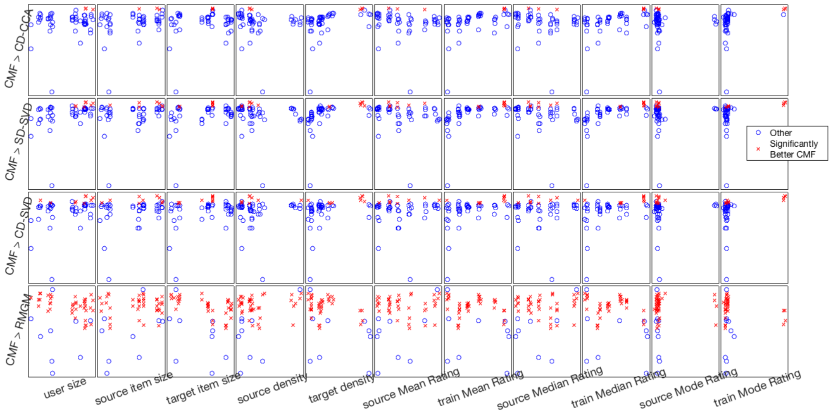


Figure 68: Scatter plot of general and central tendency statistics with the improvement ratio of CMF over other algorithms in the Supermarket dataset; red cross shows the cases in which CMF is significantly better than othor algorithms

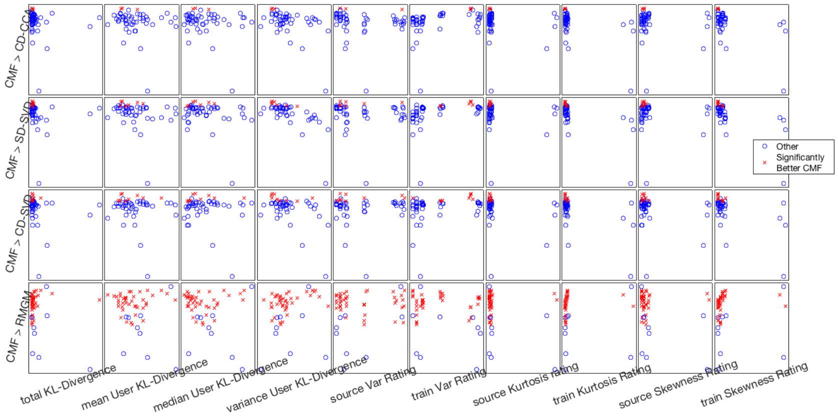


Figure 69: Scatter plot of dispersion statistics with the improvement ratio of CMF over other algorithms in the Supermarket dataset; red cross shows the cases in which CMF is significantly better than othor algorithms

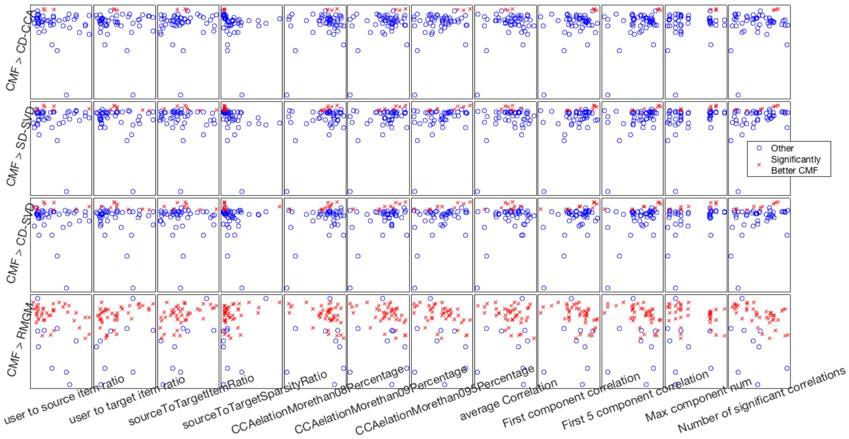


Figure 70: Scatter plot of CCA-related statistics with the improvement ratio of CMF over other algorithms in the Supermarket dataset; red cross shows the cases in which CMF is significantly better than othor algorithms

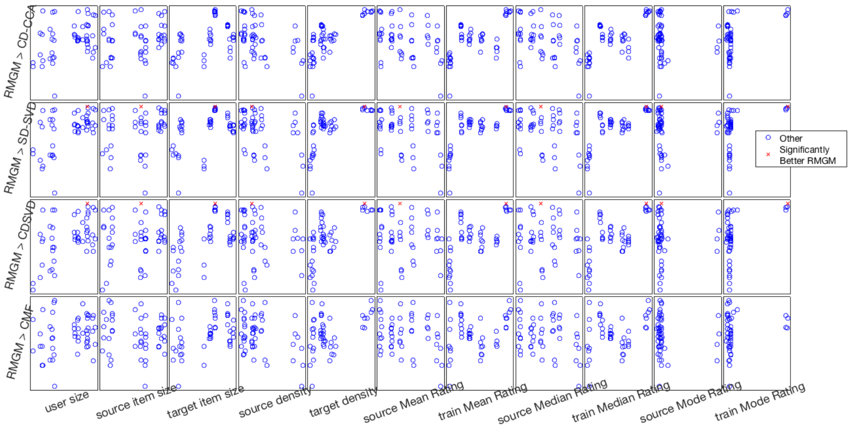


Figure 71: Scatter plot of general and central tendency statistics with the improvement ratio of RMGM over other algorithms in the Supermarket dataset; red cross shows the cases in which RMGM is significantly better than othor algorithms

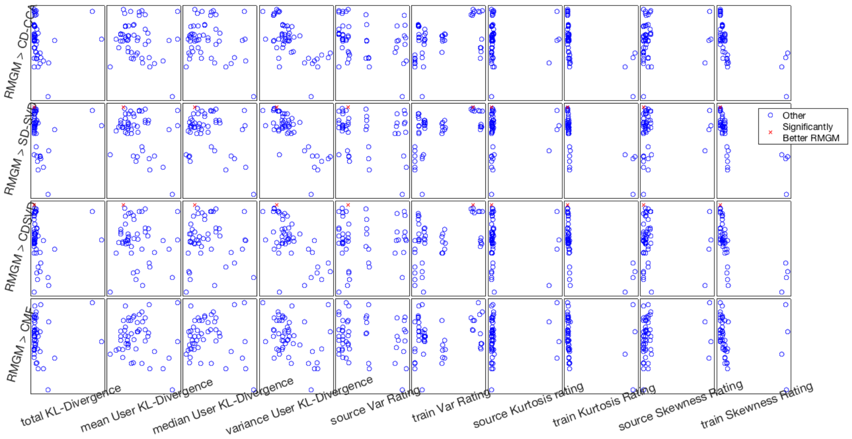


Figure 72: Scatter plot of dispersion statistics with the improvement ratio of RMGM over other algorithms in the Supermarket dataset; red cross shows the cases in which RMGM is significantly better than othor algorithms

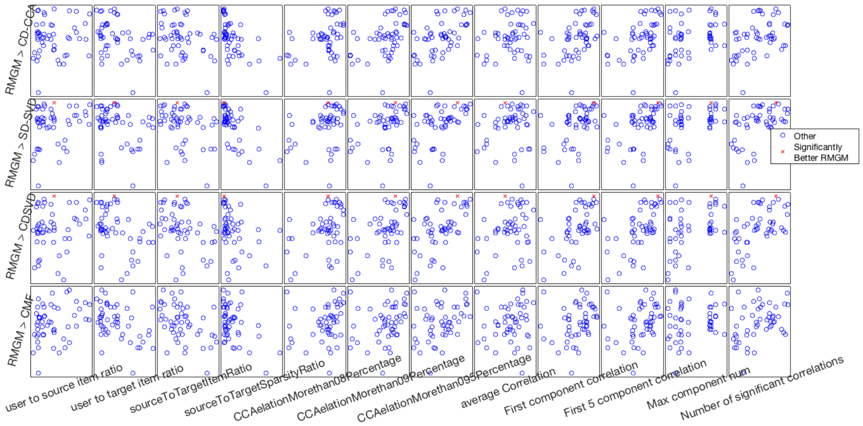


Figure 73: Scatter plot of CCA-related statistics with the improvement ratio of RMGM over other algorithms in the Supermarket dataset; red cross shows the cases in which RMGM is significantly better than othor algorithms

# APPENDIX B

# YELP DATA FIGURES AND TABLES

## B.1 DOMAIN PAIR STATISTICS FOR YELP DATASET

Table 43: Domain and domain-pair data size statistics for the Yelp dataset

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| active life | arts & entertainment | 1197 | 458 | 278 | 0.0052 | 0.0110 | 2.6135 | 4.3058 | 1.6475 | 0.4756 |
| active life | automotive | 590 | 415 | 418 | 0.0066 | 0.0041 | 1.4217 | 1.4115 | 0.9928 | 1.6035 |
| active life | beauty & spas | 864 | 442 | 528 | 0.0056 | 0.0035 | 1.9548 | 1.6364 | 0.8371 | 1.6206 |
| active life | event planning & services | 1034 | 441 | 330 | 0.0055 | 0.0058 | 2.3447 | 3.1333 | 1.3364 | 0.9473 |
| active life | food | 1891 | 501 | 1505 | 0.0040 | 0.0045 | 3.7745 | 1.2565 | 0.3329 | 0.8924 |
| active life | health & medical | 471 | 371 | 322 | 0.0072 | 0.0044 | 1.2695 | 1.4627 | 1.1522 | 1.6327 |
| active life | home services | 377 | 369 | 249 | 0.0082 | 0.0056 | 1.0217 | 1.5141 | 1.4819 | 1.4621 |
| active life | hotels & travel | 935 | 432 | 255 | 0.0058 | 0.0072 | 2.1644 | 3.6667 | 1.6941 | 0.8084 |
| active life | local services | 410 | 360 | 233 | 0.0082 | 0.0064 | 1.1389 | 1.7597 | 1.5451 | 1.2839 |
| active life | nightlife | 1761 | 488 | 608 | 0.0042 | 0.0102 | 3.6086 | 2.8964 | 0.8026 | 0.4140 |
| active life | pets | 343 | 338 | 184 | 0.0092 | 0.0082 | 1.0148 | 1.8641 | 1.8370 | 1.1300 |
| arts & entertainment | active life | 1197 | 278 | 458 | 0.0110 | 0.0052 | 4.3058 | 2.6135 | 0.6070 | 2.1026 |
| arts & entertainment | automotive | 645 | 261 | 431 | 0.0139 | 0.0040 | 2.4713 | 1.4965 | 0.6056 | 3.4504 |
| arts & entertainment | beauty & spas | 896 | 276 | 566 | 0.0119 | 0.0033 | 3.2464 | 1.5830 | 0.4876 | 3.5988 |
| arts & entertainment | education | 214 | 212 | 70 | 0.0233 | 0.0170 | 1.0094 | 3.0571 | 3.0286 | 1.3728 |
| arts & entertainment | event planning & services | 1215 | 282 | 351 | 0.0108 | 0.0053 | 4.3085 | 3.4615 | 0.8034 | 2.0210 |

Domain and domain-pair data size statistics for the Yelp dataset contd.

| source | target | user size | source item size | target item size | source density | target density | user to source item ratio | user to target item ratio | source to target item ratio | source to target density ratio |
|---|---|---|---|---|---|---|---|---|---|---|
| arts & entertainment | food | 2395 | 297 | 1543 | 0.0080 | 0.0041 | 8.0640 | 1.5522 | 0.1925 | 1.9552 |
| arts & entertainment | health & medical | 460 | 238 | 329 | 0.0153 | 0.0043 | 1.9328 | 1.3982 | 0.7234 | 3.5809 |
| arts & entertainment | home services | 399 | 231 | 260 | 0.0164 | 0.0055 | 1.7273 | 1.5346 | 0.8885 | 2.9656 |
| arts & entertainment | hotels & travel | 1083 | 275 | 284 | 0.0115 | 0.0066 | 3.9382 | 3.8134 | 0.9683 | 1.7535 |
| arts & entertainment | local flavor | 309 | 237 | 47 | 0.0217 | 0.0313 | 1.3038 | 6.5745 | 5.0426 | 0.6937 |
| arts & entertainment | local services | 451 | 247 | 244 | 0.0161 | 0.0061 | 1.8259 | 1.8484 | 1.0123 | 2.6313 |
| arts & entertainment | nightlife | 2910 | 295 | 621 | 0.0073 | 0.0083 | 9.8644 | 4.6860 | 0.4750 | 0.8828 |
| arts & entertainment | pets | 338 | 223 | 186 | 0.0194 | 0.0081 | 1.5157 | 1.8172 | 1.1989 | 2.3974 |
| arts & entertainment | public services & government | 284 | 238 | 70 | 0.0206 | 0.0199 | 1.1933 | 4.0571 | 3.4000 | 1.0346 |
| arts & entertainment | shopping | 1729 | 292 | 1510 | 0.0094 | 0.0030 | 5.9212 | 1.1450 | 0.1934 | 3.1490 |
| automotive | active life | 590 | 418 | 415 | 0.0041 | 0.0066 | 1.4115 | 1.4217 | 1.0072 | 0.6236 |
| automotive | arts & entertainment | 645 | 431 | 261 | 0.0040 | 0.0139 | 1.4965 | 2.4713 | 1.6513 | 0.2898 |
| automotive | beauty & spas | 506 | 393 | 456 | 0.0046 | 0.0045 | 1.2875 | 1.1096 | 0.8618 | 1.0224 |
| automotive | event planning & services | 475 | 387 | 250 | 0.0048 | 0.0089 | 1.2274 | 1.9000 | 1.5480 | 0.5352 |
| automotive | hotels & travel | 424 | 361 | 183 | 0.0053 | 0.0117 | 1.1745 | 2.3169 | 1.9727 | 0.4485 |
| automotive | nightlife | 1026 | 491 | 578 | 0.0031 | 0.0117 | 2.0896 | 1.7751 | 0.8495 | 0.2644 |
| beauty & spas | active life | 864 | 528 | 442 | 0.0035 | 0.0056 | 1.6364 | 1.9548 | 1.1946 | 0.6171 |
| beauty & spas | arts & entertainment | 896 | 566 | 276 | 0.0033 | 0.0119 | 1.5830 | 3.2464 | 2.0507 | 0.2779 |
| beauty & spas | automotive | 506 | 456 | 393 | 0.0045 | 0.0046 | 1.1096 | 1.2875 | 1.1603 | 0.9781 |
| beauty & spas | event planning & services | 1167 | 501 | 303 | 0.0033 | 0.0060 | 2.3293 | 3.8515 | 1.6535 | 0.5427 |
| beauty & spas | food | 1737 | 680 | 1451 | 0.0023 | 0.0045 | 2.5544 | 1.1971 | 0.4686 | 0.5219 |
| beauty & spas | health & medical | 547 | 451 | 336 | 0.0043 | 0.0040 | 1.2129 | 1.6280 | 1.3423 | 1.0777 |
| beauty & spas | hotels & travel | 1067 | 459 | 231 | 0.0036 | 0.0076 | 2.3246 | 4.6190 | 1.9870 | 0.4692 |
| beauty & spas | nightlife | 1576 | 661 | 599 | 0.0024 | 0.0098 | 2.3843 | 2.6311 | 1.1035 | 0.2478 |
| education | arts & entertainment | 214 | 70 | 212 | 0.0170 | 0.0233 | 3.0571 | 1.0094 | 0.3302 | 0.7285 |
| education | event planning & services | 169 | 68 | 159 | 0.0178 | 0.0193 | 2.4853 | 1.0629 | 0.4277 | 0.9254 |

Domain and domain-pair data size statistics for the Yelp dataset contd.

| source | target | user size | source item size | target item size | source density | target density | user to source item ratio | user to target item ratio | source to target item ratio | source to target density ratio |
|---|---|---|---|---|---|---|---|---|---|---|
| education | hotels & travel | 155 | 59 | 117 | 0.0207 | 0.0235 | 2.6271 | 1.3248 | 0.5043 | 0.8798 |
| education | local flavor | 67 | 41 | 40 | 0.0328 | 0.0511 | 1.6341 | 1.6750 | 1.0250 | 0.6409 |
| education | local services | 142 | 54 | 113 | 0.0223 | 0.0149 | 2.6296 | 1.2566 | 0.4779 | 1.4972 |
| education | public services & government | 64 | 42 | 47 | 0.0316 | 0.0356 | 1.5238 | 1.3617 | 0.8936 | 0.8890 |
| education | religious organizations | 19 | 15 | 12 | 0.0772 | 0.0965 | 1.2667 | 1.5833 | 1.2500 | 0.8000 |
| event planning & services | active life | 1034 | 330 | 441 | 0.0058 | 0.0055 | 3.1333 | 2.3447 | 0.7483 | 1.0556 |
| event planning & services | arts & entertainment | 1215 | 351 | 282 | 0.0053 | 0.0108 | 3.4615 | 4.3085 | 1.2447 | 0.4948 |
| event planning & services | automotive | 475 | 250 | 387 | 0.0089 | 0.0048 | 1.9000 | 1.2274 | 0.6460 | 1.8685 |
| event planning & services | beauty & spas | 1167 | 303 | 501 | 0.0060 | 0.0033 | 3.8515 | 2.3293 | 0.6048 | 1.8425 |
| event planning & services | education | 169 | 159 | 68 | 0.0193 | 0.0178 | 1.0629 | 2.4853 | 2.3382 | 1.0807 |
| event planning & services | food | 2021 | 395 | 1481 | 0.0041 | 0.0041 | 5.1165 | 1.3646 | 0.2667 | 0.9886 |
| event planning & services | health & medical | 350 | 228 | 298 | 0.0099 | 0.0051 | 1.5351 | 1.1745 | 0.7651 | 1.9457 |
| event planning & services | home services | 336 | 207 | 243 | 0.0109 | 0.0061 | 1.6232 | 1.3827 | 0.8519 | 1.7821 |
| event planning & services | hotels & travel | 4019 | 425 | 360 | 0.0032 | 0.0038 | 9.4565 | 11.1639 | 1.1806 | 0.8419 |
| event planning & services | local flavor | 227 | 196 | 47 | 0.0149 | 0.0338 | 1.1582 | 4.8298 | 4.1702 | 0.4391 |
| event planning & services | local services | 342 | 215 | 223 | 0.0113 | 0.0070 | 1.5907 | 1.5336 | 0.9641 | 1.6102 |
| event planning & services | nightlife | 2043 | 394 | 611 | 0.0041 | 0.0090 | 5.1853 | 3.3437 | 0.6448 | 0.4569 |
| event planning & services | pets | 295 | 211 | 183 | 0.0115 | 0.0086 | 1.3981 | 1.6120 | 1.1530 | 1.3336 |
| event planning & services | public services & government | 224 | 214 | 69 | 0.0137 | 0.0208 | 1.0467 | 3.2464 | 3.1014 | 0.6599 |
| financial services | professional services | 38 | 24 | 26 | 0.0493 | 0.0567 | 1.5833 | 1.4615 | 0.9231 | 0.8705 |
| financial services | public services & government | 26 | 24 | 26 | 0.0497 | 0.0577 | 1.0833 | 1.0000 | 0.9231 | 0.8611 |
| food | active life | 1891 | 1505 | 501 | 0.0045 | 0.0040 | 1.2565 | 3.7745 | 3.0040 | 1.1206 |
| food | arts & entertainment | 2395 | 1543 | 297 | 0.0041 | 0.0080 | 1.5522 | 8.0640 | 5.1953 | 0.5115 |
| food | beauty & spas | 1737 | 1451 | 680 | 0.0045 | 0.0023 | 1.1971 | 2.5544 | 2.1338 | 1.9162 |

Domain and domain-pair data size statistics for the Yelp dataset contd.

| source | target | user size | source item size | target item size | source density | target density | user to source item ratio | user to target item ratio | source to target item ratio | source to target density ratio |
|---|---|---|---|---|---|---|---|---|---|---|
| food | event planning & services | 2021 | 1481 | 395 | 0.0041 | 0.0041 | 1.3646 | 5.1165 | 3.7494 | 1.0116 |
| food | hotels & travel | 1666 | 1454 | 316 | 0.0044 | 0.0053 | 1.1458 | 5.2722 | 4.6013 | 0.8341 |
| food | nightlife | 5292 | 1573 | 632 | 0.0026 | 0.0060 | 3.3643 | 8.3734 | 2.4889 | 0.4320 |
| food | restaurants | 10383 | 1614 | 4435 | 0.0017 | 0.0022 | 6.4331 | 2.3411 | 0.3639 | 0.7820 |
| food | shopping | 3253 | 1568 | 1619 | 0.0035 | 0.0020 | 2.0746 | 2.0093 | 0.9685 | 1.7436 |
| health & medical | active life | 471 | 322 | 371 | 0.0044 | 0.0072 | 1.4627 | 1.2695 | 0.8679 | 0.6125 |
| health & medical | arts & entertainment | 460 | 329 | 238 | 0.0043 | 0.0153 | 1.3982 | 1.9328 | 1.3824 | 0.2793 |
| health & medical | beauty & spas | 547 | 336 | 451 | 0.0040 | 0.0043 | 1.6280 | 1.2129 | 0.7450 | 0.9279 |
| health & medical | event planning & services | 350 | 298 | 228 | 0.0051 | 0.0099 | 1.1745 | 1.5351 | 1.3070 | 0.5139 |
| health & medical | home services | 237 | 237 | 203 | 0.0067 | 0.0074 | 1.0000 | 1.1675 | 1.1675 | 0.9074 |
| health & medical | hotels & travel | 293 | 260 | 157 | 0.0059 | 0.0136 | 1.1269 | 1.8662 | 1.6561 | 0.4312 |
| health & medical | local services | 233 | 232 | 190 | 0.0069 | 0.0083 | 1.0043 | 1.2263 | 1.2211 | 0.8302 |
| health & medical | nightlife | 706 | 398 | 534 | 0.0033 | 0.0127 | 1.7739 | 1.3221 | 0.7453 | 0.2609 |
| home services | active life | 377 | 249 | 369 | 0.0056 | 0.0082 | 1.5141 | 1.0217 | 0.6748 | 0.6840 |
| home services | arts & entertainment | 399 | 260 | 231 | 0.0055 | 0.0164 | 1.5346 | 1.7273 | 1.1255 | 0.3372 |
| home services | event planning & services | 336 | 243 | 207 | 0.0061 | 0.0109 | 1.3827 | 1.6232 | 1.1739 | 0.5611 |
| home services | health & medical | 237 | 203 | 237 | 0.0074 | 0.0067 | 1.1675 | 1.0000 | 0.8565 | 1.1021 |
| home services | hotels & travel | 321 | 235 | 151 | 0.0063 | 0.0138 | 1.3660 | 2.1258 | 1.5563 | 0.4533 |
| home services | local services | 276 | 209 | 210 | 0.0071 | 0.0074 | 1.3206 | 1.3143 | 0.9952 | 0.9581 |
| home services | nightlife | 648 | 333 | 527 | 0.0039 | 0.0135 | 1.9459 | 1.2296 | 0.6319 | 0.2891 |
| home services | pets | 188 | 177 | 145 | 0.0088 | 0.0112 | 1.0621 | 1.2966 | 1.2207 | 0.7844 |
| home services | professional services | 135 | 95 | 50 | 0.0162 | 0.0233 | 1.4211 | 2.7000 | 1.9000 | 0.6973 |
| hotels & travel | active life | 935 | 255 | 432 | 0.0072 | 0.0058 | 3.6667 | 2.1644 | 0.5903 | 1.2370 |
| hotels & travel | arts & entertainment | 1083 | 284 | 275 | 0.0066 | 0.0115 | 3.8134 | 3.9382 | 1.0327 | 0.5703 |
| hotels & travel | automotive | 424 | 183 | 361 | 0.0117 | 0.0053 | 2.3169 | 1.1745 | 0.5069 | 2.2297 |
| hotels & travel | beauty & spas | 1067 | 231 | 459 | 0.0076 | 0.0036 | 4.6190 | 2.3246 | 0.5033 | 2.1311 |
| hotels & travel | education | 155 | 117 | 59 | 0.0235 | 0.0207 | 1.3248 | 2.6271 | 1.9831 | 1.1366 |
| hotels & travel | event planning & services | 4019 | 360 | 425 | 0.0038 | 0.0032 | 11.1639 | 9.4565 | 0.8471 | 1.1878 |
| hotels & travel | food | 1666 | 316 | 1454 | 0.0053 | 0.0044 | 5.2722 | 1.1458 | 0.2173 | 1.1989 |
| hotels & travel | health & medical | 293 | 157 | 260 | 0.0136 | 0.0059 | 1.8662 | 1.1269 | 0.6038 | 2.3192 |
| hotels & travel | home services | 321 | 151 | 235 | 0.0138 | 0.0063 | 2.1258 | 1.3660 | 0.6426 | 2.2058 |
| hotels & travel | local flavor | 223 | 147 | 43 | 0.0178 | 0.0368 | 1.5170 | 5.1860 | 3.4186 | 0.4839 |

Domain and domain-pair data size statistics for the Yelp dataset contd.

| source | target | user size | source item size | target item size | source density | target density | user to source item ratio | user to target item ratio | source to target item ratio | source to target density ratio |
|---|---|---|---|---|---|---|---|---|---|---|
| hotels & travel | local services | 303 | 154 | 209 | 0.0144 | 0.0078 | 1.9675 | 1.4498 | 0.7368 | 1.8574 |
| hotels & travel | nightlife | 1787 | 319 | 608 | 0.0052 | 0.0093 | 5.6019 | 2.9391 | 0.5247 | 0.5566 |
| hotels & travel | pets | 258 | 152 | 170 | 0.0146 | 0.0096 | 1.6974 | 1.5176 | 0.8941 | 1.5249 |
| hotels & travel | public services & government | 258 | 153 | 68 | 0.0159 | 0.0209 | 1.6863 | 3.7941 | 2.2500 | 0.7593 |
| local flavor | arts & entertainment | 309 | 47 | 237 | 0.0313 | 0.0217 | 6.5745 | 1.3038 | 0.1983 | 1.4416 |
| local flavor | education | 67 | 40 | 41 | 0.0511 | 0.0328 | 1.6750 | 1.6341 | 0.9756 | 1.5603 |
| local flavor | event planning & services | 227 | 47 | 196 | 0.0338 | 0.0149 | 4.8298 | 1.1582 | 0.2398 | 2.2775 |
| local flavor | hotels & travel | 223 | 43 | 147 | 0.0368 | 0.0178 | 5.1860 | 1.5170 | 0.2925 | 2.0664 |
| local flavor | mass media | 19 | 19 | 13 | 0.1302 | 0.1579 | 1.0000 | 1.4615 | 1.4615 | 0.8246 |
| local flavor | pets | 106 | 42 | 94 | 0.0429 | 0.0185 | 2.5238 | 1.1277 | 0.4468 | 2.3232 |
| local flavor | public services & government | 107 | 40 | 51 | 0.0416 | 0.0313 | 2.6750 | 2.0980 | 0.7843 | 1.3272 |
| local services | active life | 410 | 233 | 360 | 0.0064 | 0.0082 | 1.7597 | 1.1389 | 0.6472 | 0.7789 |
| local services | arts & entertainment | 451 | 244 | 247 | 0.0061 | 0.0161 | 1.8484 | 1.8259 | 0.9879 | 0.3800 |
| local services | education | 142 | 113 | 54 | 0.0149 | 0.0223 | 1.2566 | 2.6296 | 2.0926 | 0.6679 |
| local services | event planning & services | 342 | 223 | 215 | 0.0070 | 0.0113 | 1.5336 | 1.5907 | 1.0372 | 0.6210 |
| local services | health & medical | 233 | 190 | 232 | 0.0083 | 0.0069 | 1.2263 | 1.0043 | 0.8190 | 1.2046 |
| local services | home services | 276 | 210 | 209 | 0.0074 | 0.0071 | 1.3143 | 1.3206 | 1.0048 | 1.0438 |
| local services | hotels & travel | 303 | 209 | 154 | 0.0078 | 0.0144 | 1.4498 | 1.9675 | 1.3571 | 0.5384 |
| local services | nightlife | 667 | 273 | 539 | 0.0050 | 0.0147 | 2.4432 | 1.2375 | 0.5065 | 0.3425 |
| local services | pets | 185 | 172 | 131 | 0.0096 | 0.0120 | 1.0756 | 1.4122 | 1.3130 | 0.8063 |
| mass media | local flavor | 19 | 13 | 19 | 0.1579 | 0.1302 | 1.4615 | 1.0000 | 0.6842 | 1.2128 |
| mass media | public services & government | 18 | 13 | 18 | 0.1581 | 0.0864 | 1.3846 | 1.0000 | 0.7222 | 1.8297 |
| nightlife | active life | 1761 | 608 | 488 | 0.0102 | 0.0042 | 2.8964 | 3.6086 | 1.2459 | 2.4152 |
| nightlife | arts & entertainment | 2910 | 621 | 295 | 0.0083 | 0.0073 | 4.6860 | 9.8644 | 2.1051 | 1.1327 |
| nightlife | automotive | 1026 | 578 | 491 | 0.0117 | 0.0031 | 1.7751 | 2.0896 | 1.1772 | 3.7815 |
| nightlife | beauty & spas | 1576 | 599 | 661 | 0.0098 | 0.0024 | 2.6311 | 2.3843 | 0.9062 | 4.0358 |
| nightlife | event planning & services | 2043 | 611 | 394 | 0.0090 | 0.0041 | 3.3437 | 5.1853 | 1.5508 | 2.1885 |
| nightlife | food | 5292 | 632 | 1573 | 0.0060 | 0.0026 | 8.3734 | 3.3643 | 0.4018 | 2.3149 |
| nightlife | health & medical | 706 | 534 | 398 | 0.0127 | 0.0033 | 1.3221 | 1.7739 | 1.3417 | 3.8323 |
| nightlife | home services | 648 | 527 | 333 | 0.0135 | 0.0039 | 1.2296 | 1.9459 | 1.5826 | 3.4592 |
| nightlife | hotels & travel | 1787 | 608 | 319 | 0.0093 | 0.0052 | 2.9391 | 5.6019 | 1.9060 | 1.7966 |

Domain and domain-pair data size statistics for the Yelp dataset contd.

| source | target | user size | source item size | target item size | source density | target density | user to source item ratio | user to target item ratio | source to target item ratio | source to target density ratio |
|---|---|---|---|---|---|---|---|---|---|---|
| nightlife | local services | 667 | 539 | 273 | 0.0147 | 0.0050 | 1.2375 | 2.4432 | 1.9744 | 2.9199 |
| nightlife | pets | 559 | 535 | 215 | 0.0140 | 0.0065 | 1.0449 | 2.6000 | 2.4884 | 2.1606 |
| nightlife | restaurants | 11013 | 640 | 4396 | 0.0039 | 0.0021 | 17.2078 | 2.5052 | 0.1456 | 1.8657 |
| nightlife | shopping | 2657 | 627 | 1597 | 0.0085 | 0.0022 | 4.2376 | 1.6637 | 0.3926 | 3.8539 |
| pets | active life | 343 | 184 | 338 | 0.0082 | 0.0092 | 1.8641 | 1.0148 | 0.5444 | 0.8850 |
| pets | arts & entertainment | 338 | 186 | 223 | 0.0081 | 0.0194 | 1.8172 | 1.5157 | 0.8341 | 0.4171 |
| pets | event planning & services | 295 | 183 | 211 | 0.0086 | 0.0115 | 1.6120 | 1.3981 | 0.8673 | 0.7499 |
| pets | home services | 188 | 145 | 177 | 0.0112 | 0.0088 | 1.2966 | 1.0621 | 0.8192 | 1.2748 |
| pets | hotels & travel | 258 | 170 | 152 | 0.0096 | 0.0146 | 1.5176 | 1.6974 | 1.1184 | 0.6558 |
| pets | local flavor | 106 | 94 | 42 | 0.0185 | 0.0429 | 1.1277 | 2.5238 | 2.2381 | 0.4304 |
| pets | local services | 185 | 131 | 172 | 0.0120 | 0.0096 | 1.4122 | 1.0756 | 0.7616 | 1.2403 |
| pets | nightlife | 559 | 215 | 535 | 0.0065 | 0.0140 | 2.6000 | 1.0449 | 0.4019 | 0.4628 |
| professional services | financial services | 38 | 26 | 24 | 0.0567 | 0.0493 | 1.4615 | 1.5833 | 1.0833 | 1.1487 |
| professional services | home services | 135 | 50 | 95 | 0.0233 | 0.0162 | 2.7000 | 1.4211 | 0.5263 | 1.4341 |
| professional services | religious organizations | 9 | 9 | 8 | 0.1111 | 0.1250 | 1.0000 | 1.1250 | 1.1250 | 0.8889 |
| public services & government | arts & entertainment | 284 | 70 | 238 | 0.0199 | 0.0206 | 4.0571 | 1.1933 | 0.2941 | 0.9665 |
| public services & government | education | 64 | 47 | 42 | 0.0356 | 0.0316 | 1.3617 | 1.5238 | 1.1190 | 1.1249 |
| public services & government | event planning & services | 224 | 69 | 214 | 0.0208 | 0.0137 | 3.2464 | 1.0467 | 0.3224 | 1.5154 |
| public services & government | financial services | 26 | 26 | 24 | 0.0577 | 0.0497 | 1.0000 | 1.0833 | 1.0833 | 1.1613 |
| public services & government | hotels & travel | 258 | 68 | 153 | 0.0209 | 0.0159 | 3.7941 | 1.6863 | 0.4444 | 1.3170 |
| public services & government | local flavor | 107 | 51 | 40 | 0.0313 | 0.0416 | 2.0980 | 2.6750 | 1.2750 | 0.7535 |
| public services & government | mass media | 18 | 18 | 13 | 0.0864 | 0.1581 | 1.0000 | 1.3846 | 1.3846 | 0.5465 |
| religious organizations | education | 19 | 12 | 15 | 0.0965 | 0.0772 | 1.5833 | 1.2667 | 0.8000 | 1.2500 |
| religious organizations | professional services | 9 | 8 | 9 | 0.1250 | 0.1111 | 1.1250 | 1.0000 | 0.8889 | 1.1250 |
| restaurants | food | 10383 | 4435 | 1614 | 0.0022 | 0.0017 | 2.3411 | 6.4331 | 2.7478 | 1.2788 |
| restaurants | nightlife | 11013 | 4396 | 640 | 0.0021 | 0.0039 | 2.5052 | 17.2078 | 6.8688 | 0.5360 |
| shopping | arts & entertainment | 1729 | 1510 | 292 | 0.0030 | 0.0094 | 1.1450 | 5.9212 | 5.1712 | 0.3176 |

Domain and domain-pair data size statistics for the Yelp dataset contd.

| source | target | user size | source item size | target item size | source density | target density | user to source item ratio | user to target item ratio | source to target item ratio | source to target density ratio |
|---|---|---|---|---|---|---|---|---|---|---|
| shopping | food | 3253 | 1619 | 1568 | 0.0020 | 0.0035 | 2.0093 | 2.0746 | 1.0325 | 0.5735 |
| shopping | nightlife | 2657 | 1597 | 627 | 0.0022 | 0.0085 | 1.6637 | 4.2376 | 2.5470 | 0.2595 |

Table 44: Domain ratings central tendency and dispersion statistics for the Yelp dataset

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| active life | arts & entertainment | 3.9993 | 3.8701 | 4 | 4 | 4 | 4 | 1.1116 | 1.0776 |
| active life | automotive | 3.9902 | 3.7613 | 4 | 4 | 4 | 5 | 1.1254 | 1.8332 |
| active life | beauty & spas | 4.0450 | 4.0302 | 4 | 4 | 5 | 5 | 1.1098 | 1.3207 |
| active life | event planning & services | 4.0380 | 3.8652 | 4 | 4 | 5 | 4 | 1.0574 | 1.1496 |
| active life | food | 4.0079 | 3.9149 | 4 | 4 | 5 | 4 | 1.1644 | 1.0691 |
| active life | health & medical | 4.0184 | 4.1609 | 4 | 5 | 5 | 5 | 1.0837 | 1.5750 |
| active life | home services | 4.0026 | 3.7405 | 4 | 4 | 5 | 5 | 1.1400 | 2.0711 |
| active life | hotels & travel | 4.0414 | 3.7515 | 4 | 4 | 4 | 4 | 1.0359 | 1.2647 |
| active life | local services | 4.0173 | 3.9755 | 4 | 4 | 5 | 5 | 1.1085 | 1.5722 |
| active life | nightlife | 3.9900 | 3.6950 | 4 | 4 | 5 | 4 | 1.1866 | 1.1100 |
| active life | pets | 4.0094 | 4.3068 | 4 | 5 | 4 | 5 | 1.0617 | 1.0574 |
| arts & entertainment | active life | 3.8701 | 3.9993 | 4 | 4 | 4 | 4 | 1.0776 | 1.1116 |
| arts & entertainment | automotive | 3.8592 | 3.7469 | 4 | 4 | 4 | 5 | 1.0393 | 1.8526 |
| arts & entertainment | beauty & spas | 3.8685 | 3.9976 | 4 | 4 | 4 | 5 | 1.0753 | 1.2976 |
| arts & entertainment | education | 3.9233 | 4.0630 | 4 | 4 | 4 | 5 | 0.9865 | 1.3241 |
| arts & entertainment | event planning & services | 3.8782 | 3.8447 | 4 | 4 | 4 | 4 | 1.0284 | 1.1009 |
| arts & entertainment | food | 3.8724 | 3.9061 | 4 | 4 | 4 | 4 | 1.1205 | 1.0856 |
| arts & entertainment | health & medical | 3.8484 | 4.1005 | 4 | 5 | 4 | 5 | 1.0731 | 1.6075 |
| arts & entertainment | home services | 3.8495 | 3.6609 | 4 | 4 | 4 | 5 | 1.0513 | 2.1026 |
| arts & entertainment | hotels & travel | 3.8788 | 3.7227 | 4 | 4 | 4 | 4 | 1.0294 | 1.2784 |
| arts & entertainment | local flavor | 3.8463 | 4.0132 | 4 | 4 | 4 | 5 | 1.0136 | 1.4259 |
| arts & entertainment | local services | 3.8592 | 3.9717 | 4 | 4 | 4 | 5 | 1.0646 | 1.6072 |
| arts & entertainment | nightlife | 3.8650 | 3.7023 | 4 | 4 | 4 | 4 | 1.1625 | 1.1538 |
| arts & entertainment | pets | 3.8701 | 4.2888 | 4 | 5 | 4 | 5 | 1.0502 | 1.0956 |
| arts & entertainment | public services & government | 3.8938 | 3.8510 | 4 | 4 | 4 | 4 | 0.9830 | 1.3676 |
| arts & entertainment | shopping | 3.8935 | 3.8416 | 4 | 4 | 4 | 4 | 1.0853 | 1.2394 |
| automotive | active life | 3.7613 | 3.9902 | 4 | 4 | 5 | 4 | 1.8332 | 1.1254 |
| automotive | arts & entertainment | 3.7469 | 3.8592 | 4 | 4 | 5 | 4 | 1.8526 | 1.0393 |
| automotive | beauty & spas | 3.7654 | 4.0097 | 4 | 4 | 5 | 5 | 1.8088 | 1.3558 |

Domain ratings central tendency and dispersion statistics for the Yelp dataset contd.

| source | target | source mean rating | target mean rating | source median rating | target median rating | source mode rating | target mode rating | source var. rating | target var. rating |
|---|---|---|---|---|---|---|---|---|---|
| automotive | event planning & services | 3.7019 | 3.8878 | 4 | 4 | 5 | 4 | 1.8199 | 1.0714 |
| automotive | hotels & travel | 3.6960 | 3.7179 | 4 | 4 | 5 | 4 | 1.8243 | 1.3566 |
| automotive | nightlife | 3.7251 | 3.6823 | 4 | 4 | 5 | 4 | 2.0401 | 1.1305 |
| beauty & spas | active life | 4.0302 | 4.0450 | 4 | 4 | 5 | 5 | 1.3207 | 1.1098 |
| beauty & spas | arts & entertainment | 3.9976 | 3.8685 | 4 | 4 | 5 | 4 | 1.2976 | 1.0753 |
| beauty & spas | automotive | 4.0097 | 3.7654 | 4 | 4 | 5 | 5 | 1.3558 | 1.8088 |
| beauty & spas | event planning & services | 4.0121 | 3.9007 | 4 | 4 | 5 | 4 | 1.2963 | 1.1893 |
| beauty & spas | food | 3.9985 | 3.8960 | 4 | 4 | 5 | 4 | 1.4671 | 1.1154 |
| beauty & spas | health & medical | 3.9588 | 4.1055 | 4 | 5 | 5 | 5 | 1.5713 | 1.8046 |
| beauty & spas | hotels & travel | 4.0086 | 3.8142 | 4 | 4 | 5 | 4 | 1.2611 | 1.2851 |
| beauty & spas | nightlife | 3.9996 | 3.6975 | 4 | 4 | 5 | 4 | 1.4311 | 1.1055 |
| education | arts & entertainment | 4.0630 | 3.9233 | 4 | 4 | 5 | 4 | 1.3241 | 0.9865 |
| education | event planning & services | 4.0683 | 3.9015 | 4 | 4 | 5 | 4 | 1.2600 | 1.1025 |
| education | hotels & travel | 4.0265 | 3.8075 | 4 | 4 | 5 | 4 | 1.3238 | 1.2052 |
| education | local flavor | 4.1444 | 4.1387 | 4 | 5 | 5 | 5 | 0.9789 | 1.1203 |
| education | local services | 3.8480 | 3.7322 | 4 | 4 | 5 | 5 | 1.8238 | 1.8524 |
| education | public services & government | 4.1882 | 3.9346 | 4 | 4 | 5 | 5 | 0.7261 | 1.2504 |
| education | religious organizations | 3.6818 | 4.1364 | 4 | 5 | 5 | 5 | 1.9416 | 2.1234 |
| event planning & services | active life | 3.8652 | 4.0380 | 4 | 4 | 4 | 5 | 1.1496 | 1.0574 |
| event planning & services | arts & entertainment | 3.8447 | 3.8782 | 4 | 4 | 4 | 4 | 1.1009 | 1.0284 |
| event planning & services | automotive | 3.8878 | 3.7019 | 4 | 4 | 4 | 5 | 1.0714 | 1.8199 |
| event planning & services | beauty & spas | 3.9007 | 4.0121 | 4 | 4 | 4 | 5 | 1.1893 | 1.2963 |
| event planning & services | education | 3.9015 | 4.0683 | 4 | 4 | 4 | 5 | 1.1025 | 1.2600 |
| event planning & services | food | 3.8305 | 3.9021 | 4 | 4 | 4 | 4 | 1.2339 | 1.0697 |
| event planning & services | health & medical | 3.8935 | 4.0717 | 4 | 5 | 4 | 5 | 1.0902 | 1.6621 |
| event planning & services | home services | 3.8188 | 3.7550 | 4 | 4 | 4 | 5 | 1.2055 | 1.9479 |
| event planning & services | hotels & travel | 3.6913 | 3.6245 | 4 | 4 | 4 | 4 | 1.4393 | 1.5001 |

Domain ratings central tendency and dispersion statistics for the Yelp dataset contd.

| source | target | source mean rating | target mean rating | source median rating | target median rating | source mode rating | target mode rating | source var. rating | target var. rating |
|---|---|---|---|---|---|---|---|---|---|
| event planning & services | local flavor | 3.8971 | 4.0776 | 4 | 4 | 4 | 5 | 0.9561 | 1.2440 |
| event planning & services | local services | 3.8987 | 3.9981 | 4 | 4 | 4 | 5 | 1.1636 | 1.6229 |
| event planning & services | nightlife | 3.7981 | 3.6884 | 4 | 4 | 4 | 4 | 1.2178 | 1.0985 |
| event planning & services | pets | 3.9007 | 4.3484 | 4 | 5 | 4 | 5 | 1.1680 | 1.0680 |
| event planning & services | public services & government | 3.9439 | 3.8602 | 4 | 4 | 4 | 5 | 1.0531 | 1.4415 |
| financial services | professional services | 4.4667 | 4.4643 | 5 | 5 | 5 | 5 | 1.3909 | 1.4896 |
| financial services | public services & government | 3.9355 | 3.7436 | 4 | 4 | 5 | 4 | 1.3957 | 1.2483 |
| food | active life | 3.9149 | 4.0079 | 4 | 4 | 4 | 5 | 1.0691 | 1.1644 |
| food | arts & entertainment | 3.9061 | 3.8724 | 4 | 4 | 4 | 4 | 1.0856 | 1.1205 |
| food | beauty & spas | 3.8960 | 3.9985 | 4 | 4 | 4 | 5 | 1.1154 | 1.4671 |
| food | event planning & services | 3.9021 | 3.8305 | 4 | 4 | 4 | 4 | 1.0697 | 1.2339 |
| food | hotels & travel | 3.8938 | 3.6797 | 4 | 4 | 4 | 4 | 1.0434 | 1.3221 |
| food | nightlife | 3.9330 | 3.7179 | 4 | 4 | 4 | 4 | 1.1308 | 1.1948 |
| food | restaurants | 3.9416 | 3.7288 | 4 | 4 | 5 | 4 | 1.2375 | 1.2592 |
| food | shopping | 3.8978 | 3.8003 | 4 | 4 | 4 | 4 | 1.1573 | 1.4224 |
| health & medical | active life | 4.1609 | 4.0184 | 5 | 4 | 5 | 5 | 1.5750 | 1.0837 |
| health & medical | arts & entertainment | 4.1005 | 3.8484 | 5 | 4 | 5 | 4 | 1.6075 | 1.0731 |
| health & medical | beauty & spas | 4.1055 | 3.9588 | 5 | 4 | 5 | 5 | 1.8046 | 1.5713 |
| health & medical | event planning & services | 4.0717 | 3.8935 | 5 | 4 | 5 | 4 | 1.6621 | 1.0902 |
| health & medical | home services | 4.0693 | 3.7486 | 5 | 4 | 5 | 5 | 1.8348 | 2.0018 |
| health & medical | hotels & travel | 4.1432 | 3.7348 | 5 | 4 | 5 | 4 | 1.5086 | 1.3248 |
| health & medical | local services | 4.2102 | 4.0383 | 5 | 4 | 5 | 5 | 1.4908 | 1.4999 |
| health & medical | nightlife | 4.0892 | 3.6929 | 5 | 4 | 5 | 4 | 1.7888 | 1.1338 |
| home services | active life | 3.7405 | 4.0026 | 4 | 4 | 5 | 5 | 2.0711 | 1.1400 |
| home services | arts & entertainment | 3.6609 | 3.8495 | 4 | 4 | 5 | 4 | 2.1026 | 1.0513 |
| home services | event planning & services | 3.7550 | 3.8188 | 4 | 4 | 5 | 4 | 1.9479 | 1.2055 |
| home services | health & medical | 3.7486 | 4.0693 | 4 | 5 | 5 | 5 | 2.0018 | 1.8348 |
| home services | hotels & travel | 3.7352 | 3.6981 | 4 | 4 | 5 | 4 | 1.8936 | 1.3698 |
| home services | local services | 3.8683 | 3.9302 | 4 | 4 | 5 | 5 | 1.9826 | 1.8040 |
| home services | nightlife | 3.6726 | 3.6672 | 4 | 4 | 5 | 4 | 2.2799 | 1.1371 |

Domain ratings central tendency and dispersion statistics for the Yelp dataset contd.

| source | target | source mean rating | target mean rating | source median rating | target median rating | source mode rating | target mode rating | source var. rating | target var. rating |
|---|---|---|---|---|---|---|---|---|---|
| home services | pets | 3.7372 | 4.2778 | 4 | 5 | 5 | 5 | 2.1259 | 1.1587 |
| home services | professional services | 3.2548 | 3.2484 | 4 | 4 | 5 | 5 | 2.7705 | 3.1366 |
| hotels & travel | active life | 3.7515 | 4.0414 | 4 | 4 | 4 | 4 | 1.2647 | 1.0359 |
| hotels & travel | arts & entertainment | 3.7227 | 3.8788 | 4 | 4 | 4 | 4 | 1.2784 | 1.0294 |
| hotels & travel | automotive | 3.7179 | 3.6960 | 4 | 4 | 4 | 5 | 1.3566 | 1.8243 |
| hotels & travel | beauty & spas | 3.8142 | 4.0086 | 4 | 4 | 4 | 5 | 1.2851 | 1.2611 |
| hotels & travel | education | 3.8075 | 4.0265 | 4 | 4 | 4 | 5 | 1.2052 | 1.3238 |
| hotels & travel | event planning & services | 3.6245 | 3.6913 | 4 | 4 | 4 | 4 | 1.5001 | 1.4393 |
| hotels & travel | food | 3.6797 | 3.8938 | 4 | 4 | 4 | 4 | 1.3221 | 1.0434 |
| hotels & travel | health & medical | 3.7348 | 4.1432 | 4 | 5 | 4 | 5 | 1.3248 | 1.5086 |
| hotels & travel | home services | 3.6981 | 3.7352 | 4 | 4 | 4 | 5 | 1.3698 | 1.8936 |
| hotels & travel | local flavor | 3.7860 | 4.0567 | 4 | 4 | 4 | 5 | 1.0845 | 1.3150 |
| hotels & travel | local services | 3.7589 | 3.9817 | 4 | 4 | 4 | 5 | 1.3636 | 1.5446 |
| hotels & travel | nightlife | 3.6550 | 3.6691 | 4 | 4 | 4 | 4 | 1.3341 | 1.0996 |
| hotels & travel | pets | 3.7596 | 4.2922 | 4 | 5 | 4 | 5 | 1.2964 | 1.1787 |
| hotels & travel | public services & government | 3.7448 | 3.8229 | 4 | 4 | 4 | 5 | 1.2670 | 1.4467 |
| local flavor | arts & entertainment | 4.0132 | 3.8463 | 4 | 4 | 5 | 4 | 1.4259 | 1.0136 |
| local flavor | education | 4.1387 | 4.1444 | 5 | 4 | 5 | 5 | 1.1203 | 0.9789 |
| local flavor | event planning & services | 4.0776 | 3.8971 | 4 | 4 | 5 | 4 | 1.2440 | 0.9561 |
| local flavor | hotels & travel | 4.0567 | 3.7860 | 4 | 4 | 5 | 4 | 1.3150 | 1.0845 |
| local flavor | mass media | 4.2979 | 3.7436 | 5 | 4 | 5 | 5 | 1.0833 | 1.8799 |
| local flavor | pets | 4.0785 | 4.3533 | 4 | 5 | 5 | 5 | 1.2201 | 0.9947 |
| local flavor | public services & government | 4.1404 | 3.8772 | 4 | 4 | 5 | 4 | 1.0367 | 1.2025 |
| local services | active life | 3.9755 | 4.0173 | 4 | 4 | 5 | 5 | 1.5722 | 1.1085 |
| local services | arts & entertainment | 3.9717 | 3.8592 | 4 | 4 | 5 | 4 | 1.6072 | 1.0646 |
| local services | education | 3.7322 | 3.8480 | 4 | 4 | 5 | 5 | 1.8524 | 1.8238 |
| local services | event planning & services | 3.9981 | 3.8987 | 4 | 4 | 5 | 4 | 1.6229 | 1.1636 |
| local services | health & medical | 4.0383 | 4.2102 | 4 | 5 | 5 | 5 | 1.4999 | 1.4908 |
| local services | home services | 3.9302 | 3.8683 | 4 | 4 | 5 | 5 | 1.8040 | 1.9826 |
| local services | hotels & travel | 3.9817 | 3.7589 | 4 | 4 | 5 | 4 | 1.5446 | 1.3636 |
| local services | nightlife | 3.9771 | 3.6912 | 5 | 4 | 5 | 4 | 1.7827 | 1.0947 |
| local services | pets | 4.0554 | 4.3759 | 4 | 5 | 5 | 5 | 1.4969 | 1.0105 |
| mass media | local flavor | 3.7436 | 4.2979 | 4 | 5 | 5 | 5 | 1.8799 | 1.0833 |

Domain ratings central tendency and dispersion statistics for the Yelp dataset contd.

| source | target | source mean rating | target mean rating | source median rating | target median rating | source mode rating | target mode rating | source var. rating | target var. rating |
|---|---|---|---|---|---|---|---|---|---|
| mass media | public services & government | 3.6486 | 4.1786 | 4 | 5 | 5 | 5 | 1.6231 | 1.1892 |
| nightlife | active life | 3.6950 | 3.9900 | 4 | 4 | 4 | 5 | 1.1100 | 1.1866 |
| nightlife | arts & entertainment | 3.7023 | 3.8650 | 4 | 4 | 4 | 4 | 1.1538 | 1.1625 |
| nightlife | automotive | 3.6823 | 3.7251 | 4 | 4 | 4 | 5 | 1.1305 | 2.0401 |
| nightlife | beauty & spas | 3.6975 | 3.9996 | 4 | 4 | 4 | 5 | 1.1055 | 1.4311 |
| nightlife | event planning & services | 3.6884 | 3.7981 | 4 | 4 | 4 | 4 | 1.0985 | 1.2178 |
| nightlife | food | 3.7179 | 3.9330 | 4 | 4 | 4 | 4 | 1.1948 | 1.1308 |
| nightlife | health & medical | 3.6929 | 4.0892 | 4 | 5 | 4 | 5 | 1.1338 | 1.7888 |
| nightlife | home services | 3.6672 | 3.6726 | 4 | 4 | 4 | 5 | 1.1371 | 2.2799 |
| nightlife | hotels & travel | 3.6691 | 3.6550 | 4 | 4 | 4 | 4 | 1.0996 | 1.3341 |
| nightlife | local services | 3.6912 | 3.9771 | 4 | 5 | 4 | 5 | 1.0947 | 1.7827 |
| nightlife | pets | 3.7004 | 4.3291 | 4 | 5 | 4 | 5 | 1.0889 | 1.1980 |
| nightlife | restaurants | 3.7118 | 3.7193 | 4 | 4 | 4 | 4 | 1.3226 | 1.2607 |
| nightlife | shopping | 3.6950 | 3.8352 | 4 | 4 | 4 | 4 | 1.1530 | 1.3455 |
| pets | active life | 4.3068 | 4.0094 | 5 | 4 | 5 | 4 | 1.0574 | 1.0617 |
| pets | arts & entertainment | 4.2888 | 3.8701 | 5 | 4 | 5 | 4 | 1.0956 | 1.0502 |
| pets | event planning & services | 4.3484 | 3.9007 | 5 | 4 | 5 | 4 | 1.0680 | 1.1680 |
| pets | home services | 4.2778 | 3.7372 | 5 | 4 | 5 | 5 | 1.1587 | 2.1259 |
| pets | hotels & travel | 4.2922 | 3.7596 | 5 | 4 | 5 | 4 | 1.1787 | 1.2964 |
| pets | local flavor | 4.3533 | 4.0785 | 5 | 4 | 5 | 5 | 0.9947 | 1.2201 |
| pets | local services | 4.3759 | 4.0554 | 5 | 4 | 5 | 5 | 1.0105 | 1.4969 |
| pets | nightlife | 4.3291 | 3.7004 | 5 | 4 | 5 | 4 | 1.1980 | 1.0889 |
| professional services | financial services | 4.4643 | 4.4667 | 5 | 5 | 5 | 5 | 1.4896 | 1.3909 |
| professional services | home services | 3.2484 | 3.2548 | 4 | 4 | 5 | 5 | 3.1366 | 2.7705 |
| professional services | religious organizations | 3.5556 | 3.8889 | 4 | 4 | 4 | 5 | 2.2778 | 1.8611 |
| public services & government | arts & entertainment | 3.8510 | 3.8938 | 4 | 4 | 4 | 4 | 1.3676 | 0.9830 |
| public services & government | education | 3.9346 | 4.1882 | 4 | 4 | 5 | 5 | 1.2504 | 0.7261 |
| public services & government | event planning & services | 3.8602 | 3.9439 | 4 | 4 | 5 | 4 | 1.4415 | 1.0531 |
| public services & government | financial services | 3.7436 | 3.9355 | 4 | 4 | 4 | 5 | 1.2483 | 1.3957 |
| public services & government | hotels & travel | 3.8229 | 3.7448 | 4 | 4 | 5 | 4 | 1.4467 | 1.2670 |
| public services & government | local flavor | 3.8772 | 4.1404 | 4 | 4 | 4 | 5 | 1.2025 | 1.0367 |

Domain ratings central tendency and dispersion statistics for the Yelp dataset contd.

| source | target | source mean rating | target mean rating | source median rating | target median rating | source mode rating | target mode rating | source var. rating | target var. rating |
|---|---|---|---|---|---|---|---|---|---|
| public services & government | mass media | 4.1786 | 3.6486 | 5 | 4 | 5 | 5 | 1.1892 | 1.6231 |
| religious organizations | education | 4.1364 | 3.6818 | 5 | 4 | 5 | 5 | 2.1234 | 1.9416 |
| religious organizations | professional services | 3.8889 | 3.5556 | 4 | 4 | 5 | 4 | 1.8611 | 2.2778 |
| restaurants | food | 3.7288 | 3.9416 | 4 | 4 | 4 | 5 | 1.2592 | 1.2375 |
| restaurants | nightlife | 3.7193 | 3.7118 | 4 | 4 | 4 | 4 | 1.2607 | 1.3226 |
| shopping | arts & entertainment | 3.8416 | 3.8935 | 4 | 4 | 4 | 4 | 1.2394 | 1.0853 |
| shopping | food | 3.8003 | 3.8978 | 4 | 4 | 4 | 4 | 1.4224 | 1.1573 |
| shopping | nightlife | 3.8352 | 3.6950 | 4 | 4 | 4 | 4 | 1.3455 | 1.1530 |

Table 45: Domain and domain-pair ratings dispersion statistics for the Yelp dataset

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| active life | arts & entertainment | 0.0152 | 16.1565 | 13.4672 | 236.1682 | 3.7809 | 3.3882 | -1.1092 | -0.8987 |
| active life | automotive | 0.0658 | 19.2769 | 18.9300 | 274.5583 | 3.8610 | 2.5329 | -1.1399 | -0.8762 |
| active life | beauty & spas | 0.0132 | 15.5345 | 10.0235 | 263.4350 | 4.0319 | 3.5866 | -1.2016 | -1.1863 |
| active life | event planning & services | 0.0161 | 14.0212 | 1.3863 | 251.4691 | 3.8780 | 3.3261 | -1.1255 | -0.9247 |
| active life | food | 0.0151 | 18.5949 | 17.6753 | 189.7263 | 3.7882 | 3.3993 | -1.1510 | -0.9297 |
| active life | health & medical | 0.1289 | 14.9855 | 1.0986 | 288.1325 | 3.7984 | 4.0242 | -1.1094 | -1.4858 |
| active life | home services | 0.1171 | 16.4696 | 8.6643 | 286.6001 | 3.6460 | 2.2552 | -1.0968 | -0.8324 |
| active life | hotels & travel | 0.0390 | 15.8673 | 11.3780 | 265.3335 | 3.9831 | 3.1197 | -1.1479 | -0.8762 |
| active life | local services | 0.0435 | 16.6326 | 11.7639 | 286.5140 | 3.8040 | 3.1804 | -1.1326 | -1.1353 |
| active life | nightlife | 0.0708 | 19.4230 | 18.4273 | 191.6564 | 3.7462 | 3.0750 | -1.1399 | -0.7370 |
| active life | pets | 0.0918 | 15.0528 | 1.2040 | 282.6181 | 3.8263 | 5.5328 | -1.1130 | -1.7356 |
| arts & entertainment | active life | 0.0154 | 14.7327 | 6.0073 | 253.2545 | 3.3882 | 3.7809 | -0.8987 | -1.1092 |
| arts & entertainment | automotive | 0.1269 | 17.9118 | 17.3287 | 276.2130 | 3.2953 | 2.5456 | -0.8415 | -0.8849 |
| arts & entertainment | beauty & spas | 0.0419 | 16.4756 | 12.0712 | 266.4649 | 3.3399 | 3.4867 | -0.8834 | -1.1264 |
| arts & entertainment | education | 0.0780 | 12.9264 | 1.0986 | 285.6175 | 3.1947 | 3.8168 | -0.7985 | -1.2616 |
| arts & entertainment | event planning & services | 0.0012 | 14.5521 | 1.3863 | 263.7732 | 3.3800 | 3.3257 | -0.8694 | -0.8817 |
| arts & entertainment | food | 0.0009 | 16.6015 | 17.0104 | 192.8078 | 3.3423 | 3.3803 | -0.9130 | -0.9288 |
| arts & entertainment | health & medical | 0.2030 | 17.9341 | 17.3287 | 301.9238 | 3.3321 | 3.6804 | -0.8611 | -1.3702 |
| arts & entertainment | home services | 0.1909 | 17.6490 | 13.0267 | 289.9907 | 3.2936 | 2.1198 | -0.8362 | -0.7416 |
| arts & entertainment | hotels & travel | 0.0179 | 16.3186 | 12.0146 | 265.6131 | 3.3381 | 3.0719 | -0.8539 | -0.8507 |
| arts & entertainment | local flavor | 0.1077 | 10.2886 | 0.9163 | 230.4512 | 3.2627 | 3.3245 | -0.7973 | -1.1235 |
| arts & entertainment | local services | 0.1136 | 15.5878 | 1.3863 | 291.2517 | 3.3375 | 3.2013 | -0.8586 | -1.1459 |
| arts & entertainment | nightlife | 0.0174 | 11.8152 | 8.6219 | 161.1118 | 3.3751 | 3.0724 | -0.9471 | -0.7661 |
| arts & entertainment | pets | 0.1758 | 14.7571 | 1.6094 | 275.7562 | 3.4343 | 5.2751 | -0.8943 | -1.6796 |
| arts & entertainment | public services & government | 0.0415 | 15.6307 | 1.6094 | 283.3873 | 3.3265 | 3.3731 | -0.8128 | -1.0401 |
| arts & entertainment | shopping | 0.0043 | 15.1347 | 11.9281 | 215.6628 | 3.3628 | 3.1713 | -0.9091 | -0.9131 |
| automotive | active life | 0.0563 | 20.9979 | 23.3926 | 236.3300 | 2.5329 | 3.8610 | -0.8762 | -1.1399 |
| automotive | arts & entertainment | 0.1003 | 21.3632 | 22.9305 | 213.5785 | 2.5456 | 3.2953 | -0.8849 | -0.8415 |
| automotive | beauty & spas | 0.0244 | 18.9409 | 18.0807 | 255.8627 | 2.6056 | 3.4492 | -0.9031 | -1.1477 |

Domain and domain-pair ratings dispersion statistics for the Yelp dataset contd.

| source | target | total KL-divergenc | mean user KL-divergenc | median user KL-divergenc | variance user KL-divergenc | source Kurtosis rating | target Kurtosis rating | source skewness rating | target skewness rating |
|---|---|---|---|---|---|---|---|---|---|
| automotive | event planning & services | 0.0736 | 22.0413 | 30.9317 | 239.7236 | 2.4665 | 3.5112 | -0.8178 | -0.9544 |
| automotive | hotels & travel | 0.0515 | 20.2143 | 22.9305 | 248.7309 | 2.4529 | 3.0372 | -0.8078 | -0.8947 |
| automotive | nightlife | 0.1816 | 22.9264 | 25.4902 | 168.2070 | 2.3487 | 3.0412 | -0.8547 | -0.7307 |
| beauty & spas | active life | 0.0128 | 16.9937 | 17.3287 | 247.6592 | 3.5866 | 4.0319 | -1.1863 | -1.2016 |
| beauty & spas | arts & entertainment | 0.0412 | 19.5163 | 18.6737 | 221.5675 | 3.4867 | 3.3399 | -1.1264 | -0.8834 |
| beauty & spas | automotive | 0.0298 | 18.5620 | 17.6753 | 259.6304 | 3.4492 | 2.6056 | -1.1477 | -0.9031 |
| beauty & spas | event planning & services | 0.0189 | 11.1916 | 0.4055 | 217.6943 | 3.5301 | 3.2710 | -1.1437 | -0.9464 |
| beauty & spas | food | 0.0583 | 19.3782 | 18.5931 | 185.4296 | 3.3849 | 3.3452 | -1.1701 | -0.9312 |
| beauty & spas | health & medical | 0.0566 | 11.5541 | 0.6931 | 252.0274 | 3.1546 | 3.5436 | -1.1094 | -1.3908 |
| beauty & spas | hotels & travel | 0.0239 | 10.9762 | 0.3466 | 210.6334 | 3.5021 | 3.0313 | -1.1163 | -0.8736 |
| beauty & spas | nightlife | 0.1349 | 21.2422 | 22.7459 | 183.9480 | 3.4183 | 3.0779 | -1.1682 | -0.7427 |
| education | arts & entertainment | 0.0730 | 19.6352 | 19.8058 | 202.1654 | 3.8168 | 3.1947 | -1.2616 | -0.7985 |
| education | event planning & services | 0.0477 | 23.6094 | 26.5129 | 183.0362 | 3.7562 | 3.4474 | -1.2185 | -0.9670 |
| education | hotels & travel | 0.0667 | 21.5271 | 23.3926 | 214.1862 | 3.7994 | 3.2858 | -1.2282 | -0.9386 |
| education | local flavor | 0.0332 | 22.1399 | 35.1761 | 239.9429 | 4.0502 | 3.2817 | -1.2019 | -1.0627 |
| education | local services | 0.0079 | 11.7629 | 0.0000 | 223.2304 | 2.7110 | 2.4909 | -0.9724 | -0.8505 |
| education | public services & government | 5.6969 | 20.3068 | 31.9864 | 284.9464 | 2.2458 | 3.3293 | -0.6002 | -1.0098 |
| education | religious organizations | 0.2824 | 14.2856 | 0.6931 | 306.5837 | 2.4914 | 3.5285 | -0.8174 | -1.4696 |
| event planning & services | active life | 0.0161 | 14.8659 | 8.6351 | 246.6920 | 3.3261 | 3.8780 | -0.9247 | -1.1255 |
| event planning & services | arts & entertainment | 0.0011 | 17.0194 | 17.3287 | 236.6391 | 3.3257 | 3.3800 | -0.8817 | -0.8694 |
| event planning & services | automotive | 0.0900 | 21.2425 | 28.7067 | 259.8635 | 3.5112 | 2.4665 | -0.9544 | -0.8178 |
| event planning & services | beauty & spas | 0.0191 | 10.2032 | 0.3662 | 217.0283 | 3.2710 | 3.5301 | -0.9464 | -1.1437 |
| event planning & services | education | 0.0490 | 17.4596 | 1.7918 | 312.5803 | 3.4474 | 3.7562 | -0.9670 | -1.2185 |
| event planning & services | food | 0.0046 | 18.4615 | 17.6753 | 197.3122 | 3.1514 | 3.3826 | -0.8903 | -0.9144 |
| event planning & services | health & medical | 0.1660 | 20.2700 | 35.3505 | 292.3892 | 3.4032 | 3.5030 | -0.9178 | -1.3150 |
| event planning & services | home services | 0.1145 | 18.5833 | 17.6753 | 291.2655 | 3.3556 | 2.3086 | -0.9529 | -0.8247 |
| event planning & services | hotels & travel | 0.0015 | 1.8845 | 0.0000 | 40.7132 | 2.7269 | 2.5759 | -0.7773 | -0.7143 |

Domain and domain-pair ratings dispersion statistics for the Yelp dataset contd.

| source | target | total KL-divergenc | mean user KL-divergenc | median user KL-divergenc | variance user KL-divergenc | source Kurtosis rating | target Kurtosis rating | source skewness rating | target skewness rating |
|---|---|---|---|---|---|---|---|---|---|
| event planning & services | local flavor | 0.0876 | 17.6478 | 17.3287 | 279.2561 | 3.2829 | 3.7287 | -0.8054 | -1.1997 |
| event planning & services | local services | 0.0876 | 17.8281 | 17.5020 | 287.7133 | 3.4911 | 3.1969 | -1.0053 | -1.1628 |
| event planning & services | nightlife | 0.0165 | 19.6571 | 19.7888 | 204.4376 | 3.0832 | 3.0659 | -0.8441 | -0.7244 |
| event planning & services | pets | 0.1697 | 16.8326 | 11.3780 | 287.8458 | 3.4311 | 5.7889 | -1.0013 | -1.8249 |
| event planning & services | public services & government | 0.0256 | 17.4523 | 17.3287 | 284.0684 | 3.6481 | 3.2124 | -1.0023 | -1.0279 |
| financial services | professional services | 1.9592 | 5.4405 | 0.0000 | 155.8866 | 7.0330 | 5.8000 | -2.3154 | -2.0961 |
| financial services | public services & government | 0.0639 | 29.5836 | 36.0437 | 178.2565 | 3.4489 | 3.6884 | -1.1074 | -1.0867 |
| food | active life | 0.0153 | 12.3180 | 1.0986 | 254.4170 | 3.3993 | 3.7882 | -0.9297 | -1.1510 |
| food | arts & entertainment | 0.0010 | 11.2891 | 0.9163 | 235.2941 | 3.3803 | 3.3423 | -0.9288 | -0.9130 |
| food | beauty & spas | 0.0599 | 12.6098 | 1.0986 | 259.9170 | 3.3452 | 3.3849 | -0.9312 | -1.1701 |
| food | event planning & services | 0.0051 | 12.2012 | 1.0986 | 256.7831 | 3.3826 | 3.1514 | -0.9144 | -0.8903 |
| food | hotels & travel | 0.0263 | 15.5777 | 1.7918 | 279.3756 | 3.3474 | 2.9343 | -0.8863 | -0.8064 |
| food | nightlife | 0.0252 | 13.7636 | 7.5602 | 221.9204 | 3.4238 | 3.0258 | -0.9865 | -0.7833 |
| food | restaurants | 0.0271 | 13.0855 | 11.6091 | 140.9268 | 3.4003 | 2.9082 | -1.0425 | -0.7785 |
| food | shopping | 0.0113 | 10.9036 | 0.9163 | 219.2576 | 3.2974 | 2.9588 | -0.9413 | -0.9092 |
| health & medical | active life | 0.1314 | 18.2738 | 17.6753 | 242.9194 | 4.0242 | 3.7984 | -1.4858 | -1.1094 |
| health & medical | arts & entertainment | 0.2122 | 23.6098 | 27.6414 | 188.1433 | 3.6804 | 3.3321 | -1.3702 | -0.8611 |
| health & medical | beauty & spas | 0.0624 | 14.4089 | 11.3780 | 234.8590 | 3.5436 | 3.1546 | -1.3908 | -1.1094 |
| health & medical | event planning & services | 0.1684 | 23.1405 | 35.2192 | 227.2406 | 3.5030 | 3.4032 | -1.3150 | -0.9178 |
| health & medical | home services | 0.0572 | 20.0403 | 23.6236 | 278.9314 | 3.3848 | 2.2705 | -1.3282 | -0.8178 |
| health & medical | hotels & travel | 0.1820 | 23.0971 | 34.9450 | 219.6363 | 3.8944 | 3.0310 | -1.4153 | -0.8755 |
| health & medical | local services | 0.0339 | 15.7537 | 11.3780 | 270.8026 | 4.4567 | 3.4694 | -1.6018 | -1.2289 |
| health & medical | nightlife | 0.3696 | 23.5183 | 25.9930 | 156.6763 | 3.5124 | 3.0177 | -1.3691 | -0.7193 |
| home services | active life | 0.0969 | 20.6191 | 22.9305 | 224.4585 | 2.2552 | 3.6460 | -0.8324 | -1.0968 |
| home services | arts & entertainment | 0.1631 | 23.2382 | 26.4704 | 187.9973 | 2.1198 | 3.2936 | -0.7416 | -0.8362 |
| home services | event planning & services | 0.1147 | 21.5849 | 25.9930 | 238.3209 | 2.3086 | 3.3556 | -0.8247 | -0.9529 |
| home services | health & medical | 0.0511 | 19.3770 | 22.9305 | 292.3334 | 2.2705 | 3.3848 | -0.8178 | -1.3282 |
| home services | hotels & travel | 0.0754 | 21.3677 | 23.3926 | 234.7059 | 2.3427 | 2.8987 | -0.8186 | -0.8436 |
| home services | local services | 0.0037 | 14.7965 | 1.0986 | 273.8873 | 2.5516 | 2.7959 | -0.9810 | -1.0459 |
| home services | nightlife | 0.2763 | 24.3710 | 27.3668 | 156.6582 | 2.0237 | 3.0186 | -0.7453 | -0.7202 |

Domain and domain-pair ratings dispersion statistics for the Yelp dataset contd.

| source | target | total KL-divergenc | mean user KL-divergenc | median user KL-divergenc | variance user KL-divergenc | source Kurtosis rating | target Kurtosis rating | source skewness rating | target skewness rating |
|---|---|---|---|---|---|---|---|---|---|
| home services | pets | 0.0847 | 16.4521 | 14.5266 | 276.1422 | 2.2082 | 5.0859 | -0.8193 | -1.6590 |
| home services | professional services | 0.0316 | 2.0158 | 0.0000 | 65.1544 | 1.4162 | 1.2720 | -0.3138 | -0.2686 |
| hotels & travel | active life | 0.0383 | 16.9877 | 17.3287 | 252.3528 | 3.1197 | 3.9831 | -0.8762 | -1.1479 |
| hotels & travel | arts & entertainment | 0.0147 | 18.6178 | 17.6753 | 231.4209 | 3.0719 | 3.3381 | -0.8507 | -0.8539 |
| hotels & travel | automotive | 0.0517 | 19.3896 | 18.2670 | 263.9217 | 3.0372 | 2.4529 | -0.8947 | -0.8078 |
| hotels & travel | beauty & spas | 0.0247 | 9.7326 | 0.2877 | 211.5293 | 3.0313 | 3.5021 | -0.8736 | -1.1163 |
| hotels & travel | education | 0.0667 | 17.1626 | 1.6094 | 309.0994 | 3.2858 | 3.7994 | -0.9386 | -1.2282 |
| hotels & travel | event planning & services | 0.0015 | 1.6047 | 0.0000 | 36.6861 | 2.5759 | 2.7269 | -0.7143 | -0.7773 |
| hotels & travel | food | 0.0218 | 21.1512 | 22.9305 | 192.8912 | 2.9343 | 3.3474 | -0.8064 | -0.8863 |
| hotels & travel | health & medical | 0.1884 | 20.3504 | 35.3505 | 288.5156 | 3.0310 | 3.8944 | -0.8755 | -1.4153 |
| hotels & travel | home services | 0.0751 | 18.7845 | 18.0807 | 285.7877 | 2.8987 | 2.3427 | -0.8436 | -0.8186 |
| hotels & travel | local flavor | 0.1228 | 19.8497 | 18.7150 | 281.5169 | 3.4412 | 3.5307 | -0.8958 | -1.1636 |
| hotels & travel | local services | 0.0744 | 18.6095 | 18.0218 | 277.4827 | 3.0104 | 3.1434 | -0.9000 | -1.1083 |
| hotels & travel | nightlife | 0.0120 | 20.2786 | 21.3691 | 203.5635 | 2.8102 | 3.0307 | -0.7519 | -0.7073 |
| hotels & travel | pets | 0.2148 | 18.5573 | 18.0218 | 287.8745 | 3.1371 | 5.3546 | -0.9115 | -1.7387 |
| hotels & travel | public services & government | 0.0213 | 14.2976 | 1.3863 | 271.4928 | 3.0233 | 3.0492 | -0.8389 | -0.9491 |
| local flavor | arts & entertainment | 0.1030 | 17.9789 | 17.3287 | 172.8263 | 3.3245 | 3.2627 | -1.1235 | -0.7973 |
| local flavor | education | 0.0351 | 19.6478 | 36.0437 | 307.4457 | 3.2817 | 4.0502 | -1.0627 | -1.2019 |
| local flavor | event planning & services | 0.0829 | 21.1874 | 23.3926 | 217.8478 | 3.7287 | 3.2829 | -1.1997 | -0.8054 |
| local flavor | hotels & travel | 0.1156 | 22.6334 | 27.1636 | 217.8553 | 3.5307 | 3.4412 | -1.1636 | -0.8958 |
| local flavor | mass media | 0.1710 | 12.4552 | 7.1007 | 212.1137 | 4.2211 | 2.3783 | -1.4378 | -0.8315 |
| local flavor | pets | 0.0597 | 18.0972 | 17.3287 | 291.8155 | 3.5969 | 5.0360 | -1.1422 | -1.6460 |
| local flavor | public services & government | 0.0512 | 16.7442 | 17.3287 | 268.0685 | 3.9401 | 3.6412 | -1.1852 | -1.0441 |
| local services | active life | 0.0417 | 20.2114 | 20.7622 | 223.7674 | 3.1804 | 3.8040 | -1.1353 | -1.1326 |
| local services | arts & entertainment | 0.1102 | 21.3030 | 21.5530 | 193.6949 | 3.2013 | 3.3375 | -1.1459 | -0.8586 |
| local services | education | 0.0080 | 9.1866 | 0.0000 | 229.1951 | 2.4909 | 2.7110 | -0.8505 | -0.9724 |
| local services | event planning & services | 0.0879 | 21.2249 | 23.6236 | 231.2368 | 3.1969 | 3.4911 | -1.1628 | -1.0053 |
| local services | health & medical | 0.0312 | 14.9776 | 1.0986 | 279.5438 | 3.4694 | 4.4567 | -1.2289 | -1.6018 |
| local services | home services | 0.0038 | 15.2654 | 7.3551 | 268.7982 | 2.7959 | 2.5516 | -1.0459 | -0.9810 |
| local services | hotels & travel | 0.0721 | 20.7716 | 22.9305 | 238.1435 | 3.1434 | 3.0104 | -1.1083 | -0.9000 |
| local services | nightlife | 0.2591 | 22.4416 | 24.6197 | 153.4333 | 3.0581 | 3.1309 | -1.1616 | -0.7460 |
| local services | pets | 0.0476 | 14.4732 | 0.9364 | 260.7341 | 3.4832 | 6.3152 | -1.2424 | -1.9099 |

Domain and domain-pair ratings dispersion statistics for the Yelp dataset contd.

| source | target | total KL-divergenc | mean user KL-divergenc | median user KL-divergenc | variance user KL-divergenc | source Kurtosis rating | target Kurtosis rating | source skewness rating | target skewness rating |
|---|---|---|---|---|---|---|---|---|---|
| mass media | local flavor | 0.1387 | 12.5339 | 5.5567 | 217.0714 | 2.3783 | 4.2211 | -0.8315 | -1.4378 |
| mass media | public services & government | 0.1741 | 21.4680 | 26.3257 | 263.8726 | 2.0721 | 3.8321 | -0.5421 | -1.2321 |
| nightlife | active life | 0.0741 | 13.2852 | 1.3083 | 262.1594 | 3.0750 | 3.7462 | -0.7370 | -1.1399 |
| nightlife | arts & entertainment | 0.0180 | 6.4847 | 0.5233 | 153.0407 | 3.0724 | 3.3751 | -0.7661 | -0.9471 |
| nightlife | automotive | 0.1983 | 15.6377 | 1.7918 | 288.0214 | 3.0412 | 2.3487 | -0.7307 | -0.8547 |
| nightlife | beauty & spas | 0.1422 | 14.9697 | 1.6094 | 277.9625 | 3.0779 | 3.4183 | -0.7427 | -1.1682 |
| nightlife | event planning & services | 0.0174 | 14.0930 | 1.3863 | 273.4589 | 3.0659 | 3.0832 | -0.7244 | -0.8441 |
| nightlife | food | 0.0260 | 13.7288 | 7.6144 | 220.7040 | 3.0258 | 3.4238 | -0.7833 | -0.9865 |
| nightlife | health & medical | 0.3548 | 14.7964 | 1.6094 | 292.3288 | 3.0177 | 3.5124 | -0.7193 | -1.3691 |
| nightlife | home services | 0.2859 | 16.2536 | 1.9459 | 298.0849 | 3.0186 | 2.0237 | -0.7202 | -0.7453 |
| nightlife | hotels & travel | 0.0128 | 15.0443 | 1.7047 | 272.9551 | 3.0307 | 2.8102 | -0.7073 | -0.7519 |
| nightlife | local services | 0.2669 | 12.9002 | 1.3863 | 271.1494 | 3.1309 | 3.0581 | -0.7460 | -1.1616 |
| nightlife | pets | 0.3940 | 12.7763 | 1.3863 | 268.6771 | 3.1359 | 5.5664 | -0.7474 | -1.8289 |
| nightlife | restaurants | 0.0012 | 10.2608 | 8.4486 | 111.2599 | 2.9164 | 2.9072 | -0.8100 | -0.7753 |
| nightlife | shopping | 0.0337 | 14.2092 | 4.6575 | 241.7667 | 3.0274 | 3.0938 | -0.7455 | -0.9356 |
| pets | active life | 0.0951 | 18.9861 | 17.6753 | 225.0063 | 5.5328 | 3.8263 | -1.7356 | -1.1130 |
| pets | arts & entertainment | 0.1780 | 21.8856 | 22.9305 | 171.9909 | 5.2751 | 3.4343 | -1.6796 | -0.8943 |
| pets | event planning & services | 0.1706 | 19.9114 | 19.4444 | 233.8959 | 5.7889 | 3.4311 | -1.8249 | -1.0013 |
| pets | home services | 0.1091 | 16.7041 | 14.8155 | 274.5974 | 5.0859 | 2.2082 | -1.6590 | -0.8193 |
| pets | hotels & travel | 0.2091 | 21.7151 | 24.5246 | 224.5803 | 5.3546 | 3.1371 | -1.7387 | -0.9115 |
| pets | local flavor | 0.0702 | 19.3500 | 17.6753 | 268.4746 | 5.0360 | 3.5969 | -1.6460 | -1.1422 |
| pets | local services | 0.0608 | 14.7480 | 5.1836 | 258.6983 | 6.3152 | 3.4832 | -1.9099 | -1.2424 |
| pets | nightlife | 0.3926 | 22.3406 | 23.5163 | 155.9775 | 5.5664 | 3.1359 | -1.8289 | -0.7474 |
| professional services | financial services | 0.0993 | 5.9782 | 0.0000 | 158.6266 | 5.8000 | 7.0330 | -2.0961 | -2.3154 |
| professional services | home services | 0.0350 | 4.8200 | 0.0000 | 102.0316 | 1.2720 | 1.4162 | -0.2686 | -0.3138 |
| professional services | religious organizations | 8.2015 | 24.0291 | 36.0437 | 324.7862 | 1.8306 | 3.1260 | -0.5752 | -1.0457 |
| public services & government | arts & entertainment | 0.0325 | 21.6146 | 23.3207 | 187.6150 | 3.3731 | 3.3265 | -1.0401 | -0.8128 |
| public services & government | education | 0.2489 | 19.9075 | 27.4909 | 294.6180 | 3.3293 | 2.2458 | -1.0098 | -0.6002 |
| public services & government | event planning & services | 0.0213 | 21.8168 | 23.3926 | 205.9713 | 3.2124 | 3.6481 | -1.0279 | -1.0023 |
| public services & government | financial services | 0.0685 | 28.5679 | 36.0437 | 208.8340 | 3.6884 | 3.4489 | -1.0867 | -1.1074 |

Domain and domain-pair ratings dispersion statistics for the Yelp dataset contd.

| source | target | total KL-divergenc | mean user KL-divergenc | median user KL-divergenc | variance user KL-divergenc | source Kurtosis rating | target Kurtosis rating | source skewness rating | target skewness rating |
|---|---|---|---|---|---|---|---|---|---|
| public services & government | hotels & travel | 0.0208 | 18.2281 | 17.3287 | 229.4528 | 3.0492 | 3.0233 | -0.9491 | -0.8389 |
| public services & government | local flavor | 0.0508 | 15.5181 | 1.3863 | 280.8956 | 3.6412 | 3.9401 | -1.0441 | -1.1852 |
| public services & government | mass media | 0.2347 | 22.2018 | 26.4708 | 235.5339 | 3.8321 | 2.0721 | -1.2321 | -0.5421 |
| religious organizations | education | 6.7125 | 15.4954 | 0.6931 | 301.6879 | 3.5285 | 2.4914 | -1.4696 | -0.8174 |
| religious organizations | professional services | 8.2030 | 24.0291 | 36.0437 | 324.7862 | 3.1260 | 1.8306 | -1.0457 | -0.5752 |
| restaurants | food | 0.0277 | 5.3245 | 0.5952 | 128.6955 | 2.9082 | 3.4003 | -0.7785 | -1.0425 |
| restaurants | nightlife | 0.0013 | 2.2455 | 0.3964 | 52.2595 | 2.9072 | 2.9164 | -0.7753 | -0.8100 |
| shopping | arts & entertainment | 0.0039 | 13.0087 | 1.4469 | 232.0976 | 3.1713 | 3.3628 | -0.9131 | -0.9091 |
| shopping | food | 0.0096 | 14.5273 | 11.3969 | 199.4479 | 2.9588 | 3.2974 | -0.9092 | -0.9413 |
| shopping | nightlife | 0.0324 | 17.0565 | 17.3287 | 211.1177 | 3.0938 | 3.0274 | -0.9356 | -0.7455 |

Table 46: Domain-pair CCA statistics for the Yelp dataset

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| active life | arts & entertainment | 0.1315 | 0.0319 | 0.0000 | 0.4294 | 0.9476 | 0.9321 | 251 | 125 |
| active life | automotive | 0.1018 | 0.0265 | 0.0000 | 0.3907 | 0.9395 | 0.9225 | 226 | 113 |
| active life | beauty & spas | 0.0810 | 0.0211 | 0.0000 | 0.3508 | 0.9286 | 0.9161 | 284 | 143 |
| active life | event planning & services | 0.0992 | 0.0165 | 0.0000 | 0.3779 | 0.9385 | 0.9182 | 242 | 124 |
| active life | food | 0.2401 | 0.0932 | 0.0210 | 0.5351 | 0.9754 | 0.9661 | 429 | 321 |
| active life | health & medical | 0.0966 | 0.0138 | 0.0000 | 0.3806 | 0.9297 | 0.8931 | 145 | 69 |
| active life | home services | 0.0949 | 0.0146 | 0.0000 | 0.4035 | 0.9409 | 0.8984 | 137 | 66 |
| active life | hotels & travel | 0.0796 | 0.0100 | 0.0000 | 0.3559 | 0.9201 | 0.8925 | 201 | 105 |
| active life | local services | 0.1029 | 0.0147 | 0.0000 | 0.3800 | 0.9241 | 0.8985 | 136 | 66 |
| active life | nightlife | 0.1408 | 0.0453 | 0.0048 | 0.4154 | 0.9637 | 0.9477 | 419 | 222 |
| active life | pets | 0.1313 | 0.0404 | 0.0000 | 0.4007 | 0.9235 | 0.9102 | 99 | 48 |
| arts & entertainment | active life | 0.0784 | 0.0157 | 0.0000 | 0.3874 | 0.9443 | 0.9214 | 255 | 139 |
| arts & entertainment | automotive | 0.0682 | 0.0000 | 0.0000 | 0.3640 | 0.8936 | 0.8782 | 220 | 112 |
| arts & entertainment | beauty & spas | 0.0542 | 0.0042 | 0.0000 | 0.3565 | 0.9072 | 0.8862 | 240 | 134 |
| arts & entertainment | education | 0.1458 | 0.0000 | 0.0000 | 0.4812 | 0.8988 | 0.8626 | 48 | 25 |
| arts & entertainment | event planning & services | 0.0474 | 0.0000 | 0.0000 | 0.3037 | 0.8954 | 0.8753 | 253 | 115 |
| arts & entertainment | food | 0.2125 | 0.0513 | 0.0037 | 0.6021 | 0.9555 | 0.9452 | 273 | 250 |
| arts & entertainment | health & medical | 0.0556 | 0.0000 | 0.0000 | 0.2976 | 0.8994 | 0.8714 | 162 | 73 |
| arts & entertainment | home services | 0.0719 | 0.0000 | 0.0000 | 0.3637 | 0.8968 | 0.8767 | 153 | 75 |
| arts & entertainment | hotels & travel | 0.0267 | 0.0000 | 0.0000 | 0.2809 | 0.8782 | 0.8371 | 225 | 104 |
| arts & entertainment | local flavor | 0.2250 | 0.0500 | 0.0000 | 0.5569 | 0.9221 | 0.8826 | 40 | 22 |
| arts & entertainment | local services | 0.0714 | 0.0065 | 0.0000 | 0.3604 | 0.9063 | 0.8828 | 154 | 75 |
| arts & entertainment | nightlife | 0.1455 | 0.0291 | 0.0036 | 0.4970 | 0.9523 | 0.9326 | 275 | 190 |
| arts & entertainment | pets | 0.0943 | 0.0189 | 0.0000 | 0.3700 | 0.9260 | 0.8963 | 106 | 52 |
| arts & entertainment | public services & government | 0.1343 | 0.0149 | 0.0000 | 0.4996 | 0.9116 | 0.8785 | 67 | 38 |
| arts & entertainment | shopping | 0.2169 | 0.0625 | 0.0074 | 0.5776 | 0.9630 | 0.9485 | 272 | 229 |
| automotive | active life | 0.1266 | 0.0306 | 0.0000 | 0.4068 | 0.9439 | 0.9242 | 229 | 114 |
| automotive | arts & entertainment | 0.1493 | 0.0362 | 0.0000 | 0.4109 | 0.9383 | 0.9249 | 221 | 100 |
| automotive | beauty & spas | 0.1128 | 0.0205 | 0.0000 | 0.4042 | 0.9466 | 0.9166 | 195 | 106 |

Domain-pair CCA statistics for the Yelp dataset contd.

| source | target | CCA ≥ 0.80 | CCA ≥ 0.90 | CCA ≥ 0.95 | average correlation | first component correlation | first 5 components correlation | # components | # significant correlations |
|---|---|---|---|---|---|---|---|---|---|
| automotive | event planning & services | 0.1136 | 0.0114 | 0.0000 | 0.4019 | 0.9230 | 0.9037 | 176 | 85 |
| automotive | hotels & travel | 0.1259 | 0.0296 | 0.0000 | 0.4339 | 0.9316 | 0.9134 | 135 | 65 |
| automotive | nightlife | 0.1831 | 0.0710 | 0.0109 | 0.4395 | 0.9619 | 0.9551 | 366 | 194 |
| beauty & spas | active life | 0.1025 | 0.0247 | 0.0000 | 0.3693 | 0.9408 | 0.9271 | 283 | 139 |
| beauty & spas | arts & entertainment | 0.1345 | 0.0336 | 0.0000 | 0.4037 | 0.9373 | 0.9266 | 238 | 112 |
| beauty & spas | automotive | 0.1379 | 0.0197 | 0.0000 | 0.3979 | 0.9285 | 0.9084 | 203 | 101 |
| beauty & spas | event planning & services | 0.0977 | 0.0140 | 0.0000 | 0.3694 | 0.9246 | 0.9079 | 215 | 109 |
| beauty & spas | food | 0.2364 | 0.0970 | 0.0263 | 0.4964 | 0.9777 | 0.9714 | 495 | 332 |
| beauty & spas | health & medical | 0.1419 | 0.0541 | 0.0000 | 0.4240 | 0.9480 | 0.9331 | 148 | 71 |
| beauty & spas | hotels & travel | 0.0898 | 0.0120 | 0.0000 | 0.3620 | 0.9137 | 0.8926 | 167 | 87 |
| beauty & spas | nightlife | 0.1263 | 0.0450 | 0.0064 | 0.3626 | 0.9707 | 0.9551 | 467 | 215 |
| education | arts & entertainment | 0.1837 | 0.0408 | 0.0000 | 0.5303 | 0.9422 | 0.8918 | 49 | 37 |
| education | event planning & services | 0.1163 | 0.0465 | 0.0000 | 0.4580 | 0.9184 | 0.8650 | 43 | 28 |
| education | hotels & travel | 0.0513 | 0.0000 | 0.0000 | 0.4165 | 0.8930 | 0.7884 | 39 | 24 |
| education | local flavor | 0.0690 | 0.0000 | 0.0000 | 0.3288 | 0.8365 | 0.7577 | 29 | 13 |
| education | local services | 0.0606 | 0.0000 | 0.0000 | 0.4365 | 0.8823 | 0.8015 | 33 | 19 |
| education | public services & government | 0.0714 | 0.0000 | 0.0000 | 0.3544 | 0.8605 | 0.7756 | 28 | 13 |
| education | religious organizations | 0.2500 | 0.1250 | 0.1250 | 0.4531 | 0.9655 | 0.6675 | 8 | 3 |
| event planning & services | active life | 0.0816 | 0.0122 | 0.0000 | 0.3703 | 0.9349 | 0.9073 | 245 | 133 |
| event planning & services | arts & entertainment | 0.0723 | 0.0080 | 0.0000 | 0.3412 | 0.9109 | 0.8925 | 249 | 117 |
| event planning & services | automotive | 0.0726 | 0.0056 | 0.0000 | 0.3936 | 0.9071 | 0.8821 | 179 | 96 |
| event planning & services | beauty & spas | 0.0594 | 0.0046 | 0.0000 | 0.3436 | 0.9178 | 0.8842 | 219 | 122 |
| event planning & services | education | 0.0714 | 0.0000 | 0.0000 | 0.4241 | 0.8595 | 0.8189 | 42 | 22 |
| event planning & services | food | 0.1831 | 0.0640 | 0.0145 | 0.5036 | 0.9684 | 0.9598 | 344 | 261 |
| event planning & services | health & medical | 0.0873 | 0.0079 | 0.0000 | 0.3643 | 0.9031 | 0.8760 | 126 | 63 |
| event planning & services | home services | 0.0732 | 0.0163 | 0.0000 | 0.3728 | 0.9439 | 0.8870 | 123 | 62 |
| event planning & services | hotels & travel | 0.2931 | 0.0977 | 0.0086 | 0.5694 | 0.9668 | 0.9550 | 348 | 183 |

Domain-pair CCA statistics for the Yelp dataset contd.

| source | target | CCA ≥ 0.80 | CCA ≥ 0.90 | CCA ≥ 0.95 | average correlation | first component correlation | first 5 components correlation | # components | # significant correlations |
|---|---|---|---|---|---|---|---|---|---|
| event planning & services | local flavor | 0.1750 | 0.0250 | 0.0000 | 0.4942 | 0.9104 | 0.8651 | 40 | 21 |
| event planning & services | local services | 0.1157 | 0.0248 | 0.0000 | 0.4114 | 0.9292 | 0.9015 | 121 | 65 |
| event planning & services | nightlife | 0.0912 | 0.0147 | 0.0000 | 0.3786 | 0.9423 | 0.9201 | 340 | 194 |
| event planning & services | pets | 0.1222 | 0.0222 | 0.0000 | 0.3790 | 0.9405 | 0.9010 | 90 | 43 |
| event planning & services | public services & government | 0.1270 | 0.0317 | 0.0000 | 0.4558 | 0.9205 | 0.8932 | 63 | 33 |
| financial services | professional services | 0.3000 | 0.1000 | 0.0000 | 0.4328 | 0.9397 | 0.7414 | 10 | 2 |
| financial services | public services & government | 0.0714 | 0.0000 | 0.0000 | 0.3494 | 0.8242 | 0.6419 | 14 | 7 |
| food | active life | 0.2278 | 0.0839 | 0.0096 | 0.5062 | 0.9656 | 0.9564 | 417 | 222 |
| food | arts & entertainment | 0.3223 | 0.1136 | 0.0110 | 0.6227 | 0.9593 | 0.9518 | 273 | 145 |
| food | beauty & spas | 0.1777 | 0.0661 | 0.0083 | 0.4565 | 0.9629 | 0.9543 | 484 | 259 |
| food | event planning & services | 0.1780 | 0.0475 | 0.0030 | 0.4824 | 0.9600 | 0.9444 | 337 | 185 |
| food | hotels & travel | 0.1877 | 0.0578 | 0.0108 | 0.4961 | 0.9596 | 0.9505 | 277 | 145 |
| food | nightlife | 0.1922 | 0.0549 | 0.0048 | 0.5150 | 0.9633 | 0.9540 | 619 | 357 |
| food | restaurants | 0.3481 | 0.1753 | 0.0627 | 0.6302 | 0.9890 | 0.9855 | 1580 | 1291 |
| food | shopping | 0.2013 | 0.0939 | 0.0294 | 0.4377 | 0.9812 | 0.9770 | 1257 | 625 |
| health & medical | active life | 0.1074 | 0.0201 | 0.0000 | 0.4239 | 0.9378 | 0.9136 | 149 | 79 |
| health & medical | arts & entertainment | 0.1098 | 0.0305 | 0.0000 | 0.3783 | 0.9403 | 0.9206 | 164 | 75 |
| health & medical | beauty & spas | 0.1484 | 0.0516 | 0.0000 | 0.4345 | 0.9404 | 0.9298 | 155 | 86 |
| health & medical | event planning & services | 0.0880 | 0.0160 | 0.0000 | 0.3911 | 0.9144 | 0.8928 | 125 | 63 |
| health & medical | home services | 0.1149 | 0.0230 | 0.0000 | 0.4424 | 0.9421 | 0.8983 | 87 | 40 |
| health & medical | hotels & travel | 0.0808 | 0.0101 | 0.0000 | 0.4024 | 0.9053 | 0.8698 | 99 | 48 |
| health & medical | local services | 0.1014 | 0.0145 | 0.0000 | 0.3936 | 0.9225 | 0.8743 | 69 | 27 |
| health & medical | nightlife | 0.1842 | 0.0746 | 0.0175 | 0.4655 | 0.9793 | 0.9642 | 228 | 134 |
| home services | active life | 0.1216 | 0.0203 | 0.0000 | 0.4075 | 0.9407 | 0.9103 | 148 | 77 |
| home services | arts & entertainment | 0.1203 | 0.0253 | 0.0000 | 0.3899 | 0.9308 | 0.9107 | 158 | 76 |
| home services | event planning & services | 0.0952 | 0.0159 | 0.0000 | 0.3876 | 0.9183 | 0.8847 | 126 | 61 |
| home services | health & medical | 0.1222 | 0.0222 | 0.0000 | 0.4195 | 0.9155 | 0.8948 | 90 | 43 |
| home services | hotels & travel | 0.0982 | 0.0179 | 0.0000 | 0.3961 | 0.9143 | 0.8940 | 112 | 53 |

Domain-pair CCA statistics for the Yelp dataset contd.

| source | target | CCA ≥ 0.80 | CCA ≥ 0.90 | CCA ≥ 0.95 | average correlation | first component correlation | first 5 components correlation | # components | # significant correlations |
|---|---|---|---|---|---|---|---|---|---|
| home services | local services | 0.1667 | 0.0444 | 0.0000 | 0.4380 | 0.9435 | 0.9172 | 90 | 40 |
| home services | nightlife | 0.2192 | 0.0959 | 0.0183 | 0.4789 | 0.9768 | 0.9594 | 219 | 130 |
| home services | pets | 0.1127 | 0.0141 | 0.0000 | 0.3773 | 0.9078 | 0.8841 | 71 | 30 |
| home services | professional services | 0.3226 | 0.1290 | 0.0323 | 0.5927 | 0.9536 | 0.9210 | 31 | 10 |
| hotels & travel | active life | 0.0686 | 0.0049 | 0.0000 | 0.3470 | 0.9089 | 0.8894 | 204 | 114 |
| hotels & travel | arts & entertainment | 0.0437 | 0.0044 | 0.0000 | 0.3023 | 0.9083 | 0.8715 | 229 | 107 |
| hotels & travel | automotive | 0.0685 | 0.0000 | 0.0000 | 0.3812 | 0.8870 | 0.8681 | 146 | 79 |
| hotels & travel | beauty & spas | 0.0359 | 0.0000 | 0.0000 | 0.3342 | 0.8863 | 0.8388 | 167 | 100 |
| hotels & travel | education | 0.0256 | 0.0000 | 0.0000 | 0.3944 | 0.8334 | 0.7773 | 39 | 18 |
| hotels & travel | event planning & services | 0.2849 | 0.0940 | 0.0114 | 0.5591 | 0.9703 | 0.9573 | 351 | 198 |
| hotels & travel | food | 0.1748 | 0.0629 | 0.0070 | 0.4970 | 0.9614 | 0.9477 | 286 | 221 |
| hotels & travel | health & medical | 0.0891 | 0.0099 | 0.0000 | 0.3626 | 0.9063 | 0.8718 | 101 | 51 |
| hotels & travel | home services | 0.0531 | 0.0088 | 0.0000 | 0.3463 | 0.9073 | 0.8573 | 113 | 57 |
| hotels & travel | local flavor | 0.0789 | 0.0000 | 0.0000 | 0.4279 | 0.8705 | 0.8083 | 38 | 22 |
| hotels & travel | local services | 0.0796 | 0.0000 | 0.0000 | 0.3576 | 0.8848 | 0.8588 | 113 | 55 |
| hotels & travel | nightlife | 0.0821 | 0.0107 | 0.0000 | 0.3755 | 0.9371 | 0.9138 | 280 | 170 |
| hotels & travel | pets | 0.0976 | 0.0244 | 0.0000 | 0.3410 | 0.9359 | 0.8930 | 82 | 38 |
| hotels & travel | public services & government | 0.0615 | 0.0000 | 0.0000 | 0.4032 | 0.8856 | 0.8318 | 65 | 34 |
| local flavor | arts & entertainment | 0.1500 | 0.0250 | 0.0000 | 0.5438 | 0.9182 | 0.8595 | 40 | 35 |
| local flavor | education | 0.0370 | 0.0000 | 0.0000 | 0.2965 | 0.8172 | 0.6948 | 27 | 12 |
| local flavor | event planning & services | 0.0250 | 0.0000 | 0.0000 | 0.4432 | 0.8149 | 0.7551 | 40 | 31 |
| local flavor | hotels & travel | 0.0270 | 0.0000 | 0.0000 | 0.3946 | 0.8142 | 0.7368 | 37 | 27 |
| local flavor | mass media | 0.1818 | 0.0909 | 0.0000 | 0.4271 | 0.9323 | 0.7481 | 11 | 4 |
| local flavor | pets | 0.0909 | 0.0000 | 0.0000 | 0.3698 | 0.8654 | 0.8105 | 33 | 17 |
| local flavor | public services & government | 0.0333 | 0.0000 | 0.0000 | 0.4136 | 0.8316 | 0.7625 | 30 | 17 |
| local services | active life | 0.0922 | 0.0142 | 0.0000 | 0.3661 | 0.9205 | 0.8953 | 141 | 76 |
| local services | arts & entertainment | 0.1125 | 0.0188 | 0.0000 | 0.3724 | 0.9202 | 0.9043 | 160 | 77 |
| local services | education | 0.1471 | 0.0294 | 0.0000 | 0.4143 | 0.9101 | 0.8588 | 34 | 12 |
| local services | event planning & services | 0.1066 | 0.0164 | 0.0000 | 0.3914 | 0.9225 | 0.8973 | 122 | 63 |
| local services | health & medical | 0.0704 | 0.0141 | 0.0000 | 0.3566 | 0.9239 | 0.8605 | 71 | 34 |
| local services | home services | 0.1398 | 0.0538 | 0.0108 | 0.4116 | 0.9577 | 0.9285 | 93 | 44 |

Domain-pair CCA statistics for the Yelp dataset contd.

| source | target | CCA ≥ 0.80 | CCA ≥ 0.90 | CCA ≥ 0.95 | average correlation | first component correlation | first 5 components correlation | # components | # significant correlations |
|---|---|---|---|---|---|---|---|---|---|
| local services | hotels & travel | 0.0893 | 0.0000 | 0.0000 | 0.3568 | 0.8989 | 0.8788 | 112 | 52 |
| local services | nightlife | 0.2188 | 0.1042 | 0.0260 | 0.5171 | 0.9705 | 0.9600 | 192 | 130 |
| local services | pets | 0.0833 | 0.0167 | 0.0000 | 0.3528 | 0.9039 | 0.8573 | 60 | 26 |
| mass media | local flavor | 0.1818 | 0.0909 | 0.0000 | 0.4186 | 0.9151 | 0.7011 | 11 | 5 |
| mass media | public services & government | 0.2222 | 0.1111 | 0.0000 | 0.4604 | 0.9485 | 0.7399 | 9 | 4 |
| nightlife | active life | 0.0909 | 0.0172 | 0.0000 | 0.3730 | 0.9339 | 0.9222 | 407 | 206 |
| nightlife | arts & entertainment | 0.1397 | 0.0331 | 0.0000 | 0.4785 | 0.9423 | 0.9303 | 272 | 157 |
| nightlife | automotive | 0.1121 | 0.0259 | 0.0000 | 0.3947 | 0.9391 | 0.9283 | 348 | 177 |
| nightlife | beauty & spas | 0.0632 | 0.0087 | 0.0000 | 0.3154 | 0.9277 | 0.9110 | 459 | 214 |
| nightlife | event planning & services | 0.0640 | 0.0061 | 0.0000 | 0.3497 | 0.9173 | 0.8988 | 328 | 176 |
| nightlife | food | 0.1234 | 0.0321 | 0.0000 | 0.4727 | 0.9441 | 0.9371 | 624 | 427 |
| nightlife | health & medical | 0.1075 | 0.0187 | 0.0000 | 0.3950 | 0.9480 | 0.9157 | 214 | 116 |
| nightlife | home services | 0.1553 | 0.0388 | 0.0097 | 0.4559 | 0.9595 | 0.9414 | 206 | 108 |
| nightlife | hotels & travel | 0.0693 | 0.0073 | 0.0000 | 0.3522 | 0.9238 | 0.8996 | 274 | 149 |
| nightlife | local services | 0.1514 | 0.0432 | 0.0000 | 0.4629 | 0.9376 | 0.9250 | 185 | 91 |
| nightlife | pets | 0.1832 | 0.0611 | 0.0076 | 0.4652 | 0.9582 | 0.9330 | 131 | 68 |
| nightlife | restaurants | 0.4535 | 0.1953 | 0.0488 | 0.7280 | 0.9821 | 0.9748 | 635 | 634 |
| nightlife | shopping | 0.1442 | 0.0373 | 0.0049 | 0.4490 | 0.9602 | 0.9523 | 617 | 386 |
| pets | active life | 0.1132 | 0.0189 | 0.0000 | 0.4266 | 0.9260 | 0.8876 | 106 | 65 |
| pets | arts & entertainment | 0.1261 | 0.0360 | 0.0000 | 0.4327 | 0.9402 | 0.9139 | 111 | 62 |
| pets | event planning & services | 0.0851 | 0.0000 | 0.0000 | 0.3939 | 0.8913 | 0.8642 | 94 | 50 |
| pets | home services | 0.0986 | 0.0141 | 0.0000 | 0.3984 | 0.9134 | 0.8731 | 71 | 36 |
| pets | hotels & travel | 0.0805 | 0.0000 | 0.0000 | 0.3455 | 0.8870 | 0.8601 | 87 | 41 |
| pets | local flavor | 0.0606 | 0.0000 | 0.0000 | 0.3945 | 0.8866 | 0.8006 | 33 | 13 |
| pets | local services | 0.1563 | 0.0156 | 0.0000 | 0.3770 | 0.9161 | 0.8886 | 64 | 29 |
| pets | nightlife | 0.2500 | 0.0956 | 0.0221 | 0.5590 | 0.9639 | 0.9503 | 136 | 101 |
| professional services | financial services | 0.3000 | 0.1000 | 0.1000 | 0.4052 | 0.9791 | 0.7552 | 10 | 1 |
| professional services | home services | 0.2581 | 0.0645 | 0.0000 | 0.5810 | 0.9284 | 0.8919 | 31 | 19 |
| professional services | religious organizations | 0.1667 | 0.1667 | 0.1667 | 0.4368 | 0.9718 | 0.5210 | 6 | 1 |
| public services & government | arts & entertainment | 0.1493 | 0.0149 | 0.0000 | 0.5248 | 0.9089 | 0.8865 | 67 | 51 |
| public services & government | education | 0.0370 | 0.0000 | 0.0000 | 0.3266 | 0.8449 | 0.7494 | 27 | 11 |

Domain-pair CCA statistics for the Yelp dataset contd.

| source | target | CCA ≥ 0.80 | CCA ≥ 0.90 | CCA ≥ 0.95 | average correlation | first component correlation | first 5 components correlation | # components | # significant correlations |
|---|---|---|---|---|---|---|---|---|---|
| public services & government | event planning & services | 0.0462 | 0.0000 | 0.0000 | 0.4290 | 0.8748 | 0.8200 | 65 | 44 |
| public services & government | financial services | 0.1429 | 0.0000 | 0.0000 | 0.3886 | 0.8983 | 0.7530 | 14 | 5 |
| public services & government | hotels & travel | 0.0462 | 0.0000 | 0.0000 | 0.3842 | 0.8473 | 0.8075 | 65 | 38 |
| public services & government | local flavor | 0.1333 | 0.0333 | 0.0000 | 0.4135 | 0.9136 | 0.8443 | 30 | 16 |
| public services & government | mass media | 0.1111 | 0.0000 | 0.0000 | 0.4486 | 0.8498 | 0.6603 | 9 | 4 |
| religious organizations | education | 0.1429 | 0.0000 | 0.0000 | 0.3985 | 0.8799 | 0.5090 | 7 | 3 |
| religious organizations | professional services | 0.2000 | 0.0000 | 0.0000 | 0.4982 | 0.8488 | 0.4982 | 5 | 0 |
| restaurants | food | 0.3178 | 0.1544 | 0.0491 | 0.5860 | 0.9843 | 0.9803 | 1548 | 876 |
| restaurants | nightlife | 0.5039 | 0.2512 | 0.0742 | 0.7248 | 0.9823 | 0.9790 | 633 | 325 |
| shopping | arts & entertainment | 0.3371 | 0.1386 | 0.0262 | 0.6204 | 0.9653 | 0.9602 | 267 | 105 |
| shopping | food | 0.2309 | 0.1136 | 0.0373 | 0.4540 | 0.9829 | 0.9791 | 1312 | 641 |
| shopping | nightlife | 0.2579 | 0.1157 | 0.0298 | 0.5245 | 0.9759 | 0.9715 | 605 | 292 |

## B.2 ERROR OF ALGORITHMS ON DOMAIN PAIRS IN YELP DATASET

Table 47: RMSE and MAE for domain-pairs in the Yelp dataset

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| active life | arts & entertainment | 0.9597 | 1.1231 | 1.1006 | 1.2369 | 1.3288 | 0.3359 | 0.8487 | 0.8707 | 0.9501 | 0.8584 |
| active life | automotive | 1.3184 | 1.4675 | 1.4657 | 1.6373 | 2.4854 | 0.4628 | 1.1178 | 1.1234 | 1.4491 | 0.9882 |
| active life | beauty & spas | 1.1563 | 1.2795 | 1.2789 | 1.6409 | 1.1135 | 0.3920 | 2.0695 | 1.5714 | 1.4531 | 0.8540 |
| active life | event planning & services | 1.0464 | 1.1831 | 1.1786 | 1.5126 | 1.1153 | 0.3629 | 0.9300 | 0.9346 | 1.2976 | 0.8448 |
| active life | food | 1.0078 | 1.0937 | 1.1056 | 0.9600 | 0.8794 | 0.3441 | 0.8473 | 0.8446 | 0.6695 | 0.6778 |
| active life | health & medical | 1.2836 | 1.3276 | 1.3572 | 1.8555 | 1.2761 | 0.4222 | 0.8891 | 0.8811 | 1.7216 | 1.0075 |
| active life | home services | 1.3080 | 1.6100 | 1.5434 | 1.6926 | 1.2301 | 0.4724 | 1.2287 | 1.1909 | 1.5255 | 0.8620 |
| active life | hotels & travel | 1.1138 | 1.2294 | 1.1881 | 1.5058 | 1.4115 | 0.3819 | 0.9250 | 0.9416 | 1.3104 | 0.7453 |
| active life | local services | 1.3236 | 1.3931 | 1.3777 | 1.6966 | 1.8572 | 0.4470 | 1.0037 | 0.9841 | 1.5260 | 0.9790 |
| active life | nightlife | 0.9928 | 1.1116 | 1.0973 | 0.9297 | 0.9357 | 0.3451 | 0.8779 | 0.8723 | 0.6580 | 0.7257 |
| active life | pets | 1.0687 | 1.0441 | 1.0674 | 1.8683 | 1.2105 | 0.3389 | 0.7185 | 0.6934 | 1.7405 | 0.8441 |
| arts & entertainment | active life | 0.9728 | 1.0788 | 1.1296 | 1.4353 | 0.9757 | 0.3390 | 0.8296 | 0.8738 | 1.1784 | 0.7429 |
| arts & entertainment | automotive | 1.3198 | 1.4290 | 1.4036 | 1.6033 | 1.1628 | 0.4585 | 1.0684 | 1.0796 | 1.4048 | 0.9231 |
| arts & entertainment | beauty & spas | 1.1532 | 1.2839 | 1.2879 | 1.6605 | 1.1543 | 0.3884 | 1.1631 | 0.9329 | 1.4802 | 0.8927 |
| arts & entertainment | education | 1.2148 | 1.3480 | 1.2943 | 1.8419 | 0.9880 | 0.4322 | 0.9147 | 0.8962 | 1.7459 | 0.6462 |
| arts & entertainment | event planning & services | 1.0155 | 1.1087 | 1.1430 | 1.5024 | 1.8565 | 0.3494 | 0.8463 | 0.9171 | 1.2916 | 0.6768 |
| arts & entertainment | food | 1.0077 | 1.0872 | 1.1010 | 0.9701 | 0.9516 | 0.3439 | 0.8365 | 0.8628 | 0.6797 | 0.7345 |
| arts & entertainment | health & medical | 1.3595 | 1.3723 | 1.4337 | 1.8318 | 1.3540 | 0.4530 | 1.0052 | 0.9564 | 1.6956 | 0.9711 |
| arts & entertainment | home services | 1.4175 | 1.5835 | 1.5711 | 1.6272 | 1.2427 | 0.5175 | 1.2359 | 1.2388 | 1.4470 | 0.9120 |
| arts & entertainment | hotels & travel | 1.0863 | 1.1968 | 1.2052 | 1.4732 | 1.3889 | 0.3781 | 0.9093 | 0.9331 | 1.2626 | 0.7530 |
| arts & entertainment | local flavor | 0.8945 | 1.1951 | 1.2583 | 1.6829 | 0.5908 | 0.3197 | 0.9287 | 0.9279 | 1.4986 | 0.4332 |
| arts & entertainment | local services | 1.3193 | 1.3097 | 1.3792 | 1.7231 | 1.0994 | 0.4480 | 0.9821 | 0.9721 | 1.5541 | 0.8223 |
| arts & entertainment | nightlife | 0.9952 | 1.1372 | 1.1399 | 0.9860 | 0.9949 | 0.3445 | 1.9882 | 1.9217 | 0.7064 | 0.7561 |
| arts & entertainment | pets | 1.0932 | 1.1035 | 1.1543 | 1.8441 | 1.0401 | 0.3491 | 0.9999 | 0.7794 | 1.7186 | 0.7898 |
| arts & entertainment | public services & government | 1.1259 | 1.3190 | 1.3217 | 1.6354 | 0.8619 | 0.3799 | 0.9744 | 0.9464 | 1.4800 | 0.6069 |
| arts & entertainment | shopping | 1.0680 | 1.1460 | 1.1803 | 1.1220 | 1.2663 | 0.3610 | 0.8949 | 0.9107 | 0.8352 | 0.7875 |

RMSE and MAE for domain-pairs in the Yelp dataset contd.

| source | target | CCA RMSE | CD-SVD RMSE | SD-SVD RMSE | RMGM RMSE | CMF RMSE | CCA MAE | CD-SVD MAE | SD-SVD MAE | RMGM MAE | CMF MAE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| automotive | active life | 0.9996 | 1.1571 | 1.1885 | 1.4168 | 0.8863 | 0.3441 | 0.8878 | 0.9050 | 1.1592 | 0.6605 |
| automotive | arts & entertainment | 0.9818 | 1.1114 | 1.1149 | 1.1966 | 0.7677 | 0.3431 | 0.8763 | 0.8783 | 0.9179 | 0.5951 |
| automotive | beauty & spas | 1.2485 | 1.2239 | 1.2801 | 1.6929 | 1.9018 | 0.4162 | 0.9225 | 0.9203 | 1.5297 | 1.0101 |
| automotive | event planning & services | 0.9906 | 1.0752 | 1.1551 | 1.4164 | 1.5786 | 0.3412 | 0.8438 | 0.8737 | 1.1805 | 0.8015 |
| automotive | hotels & travel | 1.1315 | 1.3655 | 1.3842 | 1.4018 | 1.1693 | 0.3901 | 0.9671 | 1.0023 | 1.1756 | 0.9071 |
| automotive | nightlife | 1.0242 | 1.1533 | 1.1596 | 0.9156 | 0.8698 | 0.3561 | 1.9969 | 1.9969 | 0.6471 | 0.6715 |
| beauty & spas | active life | 1.0253 | 1.1809 | 1.1751 | 1.4339 | 1.0114 | 0.3508 | 1.9758 | 1.7631 | 1.1683 | 0.7726 |
| beauty & spas | arts & entertainment | 0.9882 | 1.2183 | 1.2092 | 1.2233 | 0.8260 | 0.3424 | 0.9216 | 0.9178 | 0.9389 | 0.6191 |
| beauty & spas | automotive | 1.3401 | 1.4506 | 1.4450 | 1.5971 | 2.1354 | 0.4648 | 1.1115 | 1.1093 | 1.4035 | 1.0305 |
| beauty & spas | event planning & services | 1.0803 | 1.1622 | 1.1666 | 1.5713 | 1.3069 | 0.3703 | 0.9011 | 0.9201 | 1.3743 | 0.7306 |
| beauty & spas | food | 1.0237 | 1.1211 | 1.1247 | 0.9808 | 0.8094 | 0.3499 | 0.8738 | 0.8799 | 0.6921 | 0.6233 |
| beauty & spas | health & medical | 1.3647 | 1.4518 | 1.4659 | 1.8508 | 1.5539 | 0.4636 | 0.9640 | 0.9591 | 1.7126 | 0.9962 |
| beauty & spas | hotels & travel | 1.0709 | 1.1799 | 1.1827 | 1.5831 | 1.3405 | 0.3713 | 0.9046 | 0.9192 | 1.4028 | 0.6019 |
| beauty & spas | nightlife | 0.9995 | 1.2656 | 1.2504 | 0.9418 | 0.8776 | 0.3454 | 0.9430 | 0.9321 | 0.6721 | 0.6799 |
| education | arts & entertainment | 0.9522 | 1.1460 | 1.1650 | 1.1095 | 0.8209 | 0.3251 | 0.9121 | 0.9139 | 0.8244 | 0.6201 |
| education | event planning & services | 1.0348 | 1.1010 | 1.0778 | 1.4302 | 1.4236 | 0.3546 | 0.8858 | 0.8786 | 1.2074 | 0.9884 |
| education | hotels & travel | 1.0487 | 1.1869 | 1.1336 | 1.4902 | 1.0684 | 0.3698 | 0.9152 | 0.9079 | 1.2820 | 0.8124 |
| education | local flavor | 0.8850 | 1.1634 | 1.1654 | 1.6383 | 0.6630 | 0.3190 | 0.8211 | 0.8554 | 1.4711 | 0.5265 |
| education | local services | 1.3724 | 1.5285 | 1.5040 | 1.6624 | 2.2257 | 0.4995 | 1.1898 | 1.2092 | 1.5194 | 1.2536 |
| education | public services & government | 1.1954 | 1.4926 | 1.4492 | 1.5820 | 1.1611 | 0.4215 | 1.0251 | 1.0277 | 1.3915 | 0.9176 |
| education | religious organizations | 1.0699 | 0.7701 | 0.7680 | 2.0679 | 1.2557 | 0.4098 | 0.6705 | 0.6135 | 2.0065 | 1.0890 |
| event planning & services | active life | 0.9357 | 1.0721 | 1.1312 | 1.4507 | 0.9875 | 0.3257 | 0.8376 | 0.8507 | 1.2009 | 0.7460 |
| event planning & services | arts & entertainment | 0.9304 | 1.0566 | 1.0751 | 1.2368 | 0.9527 | 0.3191 | 0.8175 | 0.8275 | 0.9518 | 0.6997 |
| event planning & services | automotive | 1.3165 | 1.4113 | 1.4262 | 1.5234 | 8.0580 | 0.4666 | 1.0985 | 1.1174 | 1.3181 | 1.9342 |
| event planning & services | beauty & spas | 1.1378 | 1.1782 | 1.2166 | 1.6743 | 1.0954 | 0.3891 | 0.9093 | 0.9258 | 1.5055 | 0.8018 |
| event planning & services | education | 1.1115 | 1.2287 | 1.2475 | 1.7908 | 1.0777 | 0.3921 | 0.8578 | 0.9421 | 1.6730 | 0.8289 |
| event planning & services | food | 1.0024 | 1.0895 | 1.1083 | 0.9723 | 0.9266 | 0.3417 | 0.8403 | 0.8502 | 0.6820 | 0.7137 |
| event planning & services | health & medical | 1.3762 | 1.3025 | 1.3511 | 1.7959 | 1.1360 | 0.4576 | 0.9651 | 0.9363 | 1.6528 | 0.9165 |

RMSE and MAE for domain-pairs in the Yelp dataset contd.

| source | target | CCA RMSE | CD-SVD RMSE | SD-SVD RMSE | RMGM RMSE | CMF RMSE | CCA MAE | CD-SVD MAE | SD-SVD MAE | RMGM MAE | CMF MAE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| event planning & services | home services | 1.3984 | 1.6203 | 1.6012 | 1.6774 | 2.1187 | 0.5038 | 2.1060 | 1.4656 | 1.5157 | 1.1500 |
| event planning & services | hotels & travel | 1.1037 | 1.2480 | 1.2681 | 1.5874 | 1.0394 | 0.3916 | 0.9848 | 0.9911 | 1.4298 | 0.7247 |
| event planning & services | local flavor | 0.8565 | 1.1704 | 1.2161 | 1.6782 | 0.8818 | 0.3007 | 0.9029 | 0.9399 | 1.5063 | 0.5103 |
| event planning & services | local services | 1.2664 | 1.2694 | 1.2943 | 1.7359 | 1.2399 | 0.4369 | 2.0979 | 1.5720 | 1.5624 | 0.9150 |
| event planning & services | nightlife | 0.9899 | 1.0998 | 1.1147 | 0.9538 | 1.0406 | 0.3434 | 0.8597 | 0.8698 | 0.6786 | 0.8181 |
| event planning & services | pets | 1.0151 | 1.0471 | 1.0991 | 1.8254 | 1.0593 | 0.3249 | 0.7300 | 0.6470 | 1.6696 | 0.7791 |
| event planning & services | public services & government | 1.1623 | 1.3601 | 1.3871 | 1.6277 | 0.9349 | 0.3839 | 0.9527 | 1.0119 | 1.4531 | 0.6975 |
| financial services | professional services | 0.9415 | 0.8848 | 0.8888 | 2.0885 | 1.0712 | 0.3225 | 0.5372 | 0.5257 | 2.0268 | 0.7155 |
| financial services | public services & government | nan | 1.6876 | 1.6554 | nan | nan | nan | 0.9791 | 0.9047 | nan | nan |
| food | active life | 0.9906 | 1.1035 | 1.1493 | 1.5095 | 0.9797 | 0.3412 | 0.8376 | 0.8774 | 1.2667 | 0.7502 |
| food | arts & entertainment | 0.9550 | 1.0437 | 1.0893 | 1.3808 | 1.1854 | 0.3349 | 0.8202 | 0.8454 | 1.1175 | 0.6539 |
| food | beauty & spas | 1.2189 | 1.2276 | 1.3159 | 1.6846 | 0.9998 | 0.4132 | 0.9342 | 0.9526 | 1.5079 | 0.7666 |
| food | event planning & services | 1.0420 | 1.1497 | 1.1830 | 1.5642 | 1.3116 | 0.3618 | 0.8841 | 0.9236 | 1.3784 | 0.6785 |
| food | hotels & travel | 1.0849 | 1.2175 | 1.2226 | 1.5144 | 1.3180 | 0.3755 | 1.5607 | 0.9761 | 1.3320 | 0.6764 |
| food | nightlife | 1.0104 | 1.2556 | 1.2489 | 1.1407 | 3.2806 | 0.3502 | 2.0056 | 2.0056 | 0.8562 | 0.8894 |
| food | restaurants | 1.0740 | 1.1647 | 1.1635 | 0.8048 | 1.3621 | 0.3703 | 0.9146 | 0.9153 | 0.5598 | 0.8172 |
| food | shopping | 1.1405 | 1.2257 | 1.2485 | 1.2262 | 1.0591 | 0.3897 | 0.9387 | 0.9662 | 0.9389 | 0.8329 |
| health & medical | active life | 1.0203 | 1.1825 | 1.1565 | 1.4163 | 0.9302 | 0.3504 | 0.8957 | 0.8836 | 1.1538 | 0.6826 |
| health & medical | arts & entertainment | 0.9822 | 1.2633 | 1.2151 | 1.2021 | 0.7332 | 0.3424 | 0.8894 | 0.8934 | 0.9205 | 0.5596 |
| health & medical | beauty & spas | 1.3063 | 1.3298 | 1.3365 | 1.6416 | 1.1730 | 0.4411 | 0.9745 | 0.9749 | 1.4418 | 0.9011 |
| health & medical | event planning & services | 1.0649 | 1.1822 | 1.2052 | 1.5025 | 0.9656 | 0.3688 | 0.9206 | 0.9137 | 1.2860 | 0.7029 |
| health & medical | home services | 1.3740 | 1.5394 | 1.4995 | 1.6978 | 1.3562 | 0.5064 | 1.1355 | 1.2148 | 1.5394 | 1.1272 |
| health & medical | hotels & travel | 1.1238 | 1.2756 | 1.1767 | 1.5249 | 1.8405 | 0.3880 | 0.9887 | 0.9459 | 1.3406 | 0.8000 |
| health & medical | local services | 1.2814 | 1.3454 | 1.3898 | 1.8110 | 1.3223 | 0.4253 | 0.9428 | 0.9465 | 1.6796 | 1.0612 |
| health & medical | nightlife | 1.0411 | 1.1630 | 1.1631 | 0.9345 | 0.8119 | 0.3621 | 0.9730 | 0.9811 | 0.6623 | 0.6330 |
| home services | active life | 1.0277 | 1.1443 | 1.1663 | 1.3820 | 0.9898 | 0.3538 | 0.8729 | 0.9118 | 1.1136 | 0.7253 |
| home services | arts & entertainment | 0.9692 | 1.0846 | 1.1060 | 1.2233 | 0.7805 | 0.3338 | 0.8783 | 0.8867 | 0.9412 | 0.6050 |
| home services | event planning & services | 1.0830 | 1.2416 | 1.3295 | 1.4366 | 1.2280 | 0.3726 | 1.8063 | 1.5279 | 1.2117 | 0.7195 |

RMSE and MAE for domain-pairs in the Yelp dataset contd.

| source | target | CCA RMSE | CD-SVD RMSE | SD-SVD RMSE | RMGM RMSE | CMF RMSE | CCA MAE | CD-SVD MAE | SD-SVD MAE | RMGM MAE | CMF MAE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| home services | health & medical | 1.4840 | 1.4120 | 1.5547 | 1.7915 | 1.4781 | 0.5105 | 1.0289 | 1.0437 | 1.6400 | 1.2528 |
| home services | hotels & travel | 1.1398 | 1.2246 | 1.2311 | 1.4717 | 1.2088 | 0.3903 | 0.9801 | 0.9547 | 1.2713 | 0.7916 |
| home services | local services | 1.2935 | 1.4292 | 1.4325 | 1.7531 | 1.3400 | 0.4558 | 1.0268 | 1.0024 | 1.6058 | 1.0669 |
| home services | nightlife | 1.0663 | 1.2618 | 1.2695 | 0.8826 | 0.8670 | 0.3700 | 1.9864 | 1.9864 | 0.6277 | 0.6764 |
| home services | pets | 1.3037 | 1.2323 | 1.2445 | 1.7856 | 1.1222 | 0.4231 | 0.8051 | 0.8455 | 1.6413 | 0.8447 |
| home services | professional services | 1.6791 | 1.8771 | 1.8859 | 1.6276 | 0.9716 | 0.6612 | 1.5792 | 1.6054 | 1.4349 | 0.7811 |
| hotels & travel | active life | 0.9886 | 1.0815 | 1.1608 | 1.4026 | 1.8883 | 0.3394 | 0.8497 | 0.8746 | 1.1386 | 0.7242 |
| hotels & travel | arts & entertainment | 0.9386 | 1.0506 | 1.1052 | 1.2375 | 0.9457 | 0.3288 | 0.8366 | 0.8651 | 0.9514 | 0.7177 |
| hotels & travel | automotive | 1.3212 | 1.5675 | 1.5734 | 1.5820 | 1.6153 | 0.4677 | 1.1411 | 1.1465 | 1.4065 | 1.1032 |
| hotels & travel | beauty & spas | 1.1305 | 1.1480 | 1.2324 | 1.6805 | 1.7435 | 0.3813 | 0.8874 | 0.9249 | 1.5143 | 0.8541 |
| hotels & travel | education | 1.1876 | 1.2162 | 1.2739 | 1.7652 | 0.8588 | 0.4042 | 0.9402 | 0.9660 | 1.6323 | 0.6072 |
| hotels & travel | event planning & services | 1.0998 | 1.2240 | 1.2563 | 1.5823 | 1.0875 | 0.3896 | 0.9632 | 0.9971 | 1.4172 | 0.7549 |
| hotels & travel | food | 0.9999 | 1.0888 | 1.1162 | 0.9703 | 0.8995 | 0.3411 | 1.5087 | 0.9003 | 0.6838 | 0.7049 |
| hotels & travel | health & medical | 1.3278 | 1.3261 | 1.4130 | 1.7767 | 1.3141 | 0.4494 | 0.9663 | 0.9129 | 1.6211 | 0.9531 |
| hotels & travel | home services | 1.3209 | 1.4280 | 1.4987 | 1.6620 | 1.1125 | 0.4770 | 1.1381 | 1.1182 | 1.5023 | 0.9059 |
| hotels & travel | local flavor | 0.9177 | 1.2108 | 1.2359 | 1.6659 | 0.8704 | 0.3219 | 0.9094 | 0.9218 | 1.4946 | 0.6184 |
| hotels & travel | local services | 1.2533 | 1.2744 | 1.3401 | 1.6809 | 1.6728 | 0.4171 | 0.9617 | 0.9795 | 1.5075 | 0.9263 |
| hotels & travel | nightlife | 0.9870 | 1.0796 | 1.0942 | 0.9378 | 1.0745 | 0.3428 | 0.8468 | 0.8753 | 0.6681 | 0.8456 |
| hotels & travel | pets | 1.1931 | 1.1974 | 1.2243 | 1.8269 | 1.1148 | 0.3846 | 0.9249 | 0.7960 | 1.6776 | 0.8225 |
| hotels & travel | public services & government | 1.1799 | 1.2765 | 1.2888 | 1.6328 | 0.8503 | 0.4097 | 1.1699 | 1.0968 | 1.4700 | 0.6573 |
| local flavor | arts & entertainment | 0.9485 | 1.1059 | 1.0835 | 1.0405 | 0.8045 | 0.3303 | 0.8427 | 0.8604 | 0.7526 | 0.6184 |
| local flavor | education | 1.0328 | 1.1061 | 1.0632 | 1.8235 | 0.8217 | 0.3592 | 0.7563 | 0.7247 | 1.7131 | 0.6400 |
| local flavor | event planning & services | 0.9466 | 1.1056 | 1.1101 | 1.4569 | 1.0293 | 0.3246 | 0.9087 | 0.8748 | 1.2408 | 0.7584 |
| local flavor | hotels & travel | 1.0298 | 1.1692 | 1.1343 | 1.4457 | 1.0220 | 0.3510 | 0.8900 | 0.9495 | 1.2527 | 0.7683 |
| local flavor | mass media | 1.2381 | 1.2613 | 1.4176 | 1.5768 | 1.2267 | 0.4481 | 0.9703 | 0.9910 | 1.3894 | 1.0937 |
| local flavor | pets | 0.9813 | 1.0773 | 1.0875 | 1.8641 | 1.6895 | 0.3351 | 0.6566 | 0.6965 | 1.7263 | 0.8933 |
| local flavor | public services & government | 1.0544 | 1.2767 | 1.2206 | 1.6984 | 1.7611 | 0.3504 | 0.9610 | 0.9283 | 1.5803 | 0.9958 |
| local services | active life | 1.0148 | 1.1265 | 1.1729 | 1.3740 | 0.9769 | 0.3514 | 0.8711 | 0.8966 | 1.1060 | 0.7229 |
| local services | arts & entertainment | 0.9910 | 1.1075 | 1.1108 | 1.1786 | 0.7913 | 0.3410 | 0.8929 | 0.8903 | 0.8923 | 0.6088 |
| local services | education | 1.3073 | 1.3845 | 1.3757 | 1.7335 | 0.8844 | 0.4726 | 1.0261 | 1.0194 | 1.5851 | 0.6842 |
| local services | event planning & services | 1.0842 | 1.2030 | 1.1857 | 1.4705 | 0.8957 | 0.3697 | 1.9847 | 1.9816 | 1.2379 | 0.6633 |
| local services | health & medical | 1.2779 | 1.2585 | 1.2985 | 1.8968 | 1.0867 | 0.4244 | 0.7883 | 0.8142 | 1.7758 | 0.8823 |

RMSE and MAE for domain-pairs in the Yelp dataset contd.

| source | target | CCA RMSE | CD-SVD RMSE | SD-SVD RMSE | RMGM RMSE | CMF RMSE | CCA MAE | CD-SVD MAE | SD-SVD MAE | RMGM MAE | CMF MAE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| local services | home services | 1.3550 | 1.5605 | 1.5737 | 1.7302 | 2.9252 | 0.4758 | 1.2003 | 1.1822 | 1.5659 | 1.3186 |
| local services | hotels & travel | 1.1527 | 1.3120 | 1.2582 | 1.5036 | 7.0090 | 0.3987 | 1.0110 | 0.9963 | 1.3063 | 1.3024 |
| local services | nightlife | 1.0143 | 1.1292 | 1.1093 | 0.8861 | 0.8213 | 0.3533 | 0.8781 | 0.8788 | 0.6235 | 0.6432 |
| local services | pets | 1.0954 | 1.1223 | 1.1672 | 1.8203 | 1.5735 | 0.3623 | 0.7068 | 0.6961 | 1.6562 | 0.9083 |
| mass media | local flavor | 1.0835 | 1.0144 | 1.0363 | 1.6173 | 1.1060 | 0.4069 | 0.8843 | 0.7022 | 1.3986 | 0.9527 |
| mass media | public services & government | 0.9660 | 1.2034 | 1.4046 | 1.8621 | 1.0068 | 0.3469 | 1.2600 | 1.0851 | 1.8125 | 0.8803 |
| nightlife | active life | 0.9911 | 1.0878 | 1.1610 | 1.4749 | 0.9275 | 0.3435 | 0.8417 | 0.8970 | 1.2333 | 0.6990 |
| nightlife | arts & entertainment | 0.9894 | 1.1139 | 1.1453 | 1.3915 | 1.1792 | 0.3452 | 2.0129 | 1.8353 | 1.1329 | 0.5548 |
| nightlife | automotive | 1.3588 | 1.4917 | 1.5197 | 1.6415 | 1.0218 | 0.4698 | 2.0137 | 1.7052 | 1.4543 | 0.7873 |
| nightlife | beauty & spas | 1.2058 | 1.3708 | 1.3363 | 1.6860 | 1.0331 | 0.4084 | 1.0049 | 0.9539 | 1.5083 | 0.7810 |
| nightlife | event planning & services | 1.0489 | 1.1270 | 1.1893 | 1.5610 | 1.6780 | 0.3656 | 0.8700 | 0.9332 | 1.3831 | 0.5992 |
| nightlife | food | 1.0124 | 1.1921 | 1.2323 | 1.1348 | 1.2246 | 0.3475 | 2.0324 | 2.0312 | 0.8314 | 0.8717 |
| nightlife | health & medical | 1.3849 | 1.4264 | 1.4528 | 1.8264 | 1.1645 | 0.4610 | 1.1259 | 0.9828 | 1.6796 | 0.8691 |
| nightlife | home services | 1.4286 | 1.7171 | 1.6776 | 1.6989 | 1.5219 | 0.5163 | 2.1176 | 1.6466 | 1.5328 | 1.0140 |
| nightlife | hotels & travel | 1.0775 | 1.1701 | 1.1969 | 1.5088 | 3.9908 | 0.3747 | 0.9096 | 0.9678 | 1.3275 | 0.7230 |
| nightlife | local services | 1.3294 | 1.3387 | 1.4111 | 1.7464 | 1.4236 | 0.4489 | 1.0453 | 1.0317 | 1.5909 | 0.7823 |
| nightlife | pets | 1.0737 | 1.1745 | 1.1546 | 1.8809 | 0.9330 | 0.3479 | 0.8909 | 0.6839 | 1.7501 | 0.6781 |
| nightlife | restaurants | 1.0707 | 1.1444 | 1.1612 | 0.7992 | 1.2232 | 0.3724 | 0.8991 | 0.9125 | 0.5578 | 0.7780 |
| nightlife | shopping | 1.1062 | 1.1649 | 1.2185 | 1.2009 | 1.1053 | 0.3747 | 0.9112 | 0.9457 | 0.9156 | 0.8565 |
| pets | active life | 1.0193 | 1.1510 | 1.1466 | 1.3749 | 1.2093 | 0.3478 | 0.8810 | 0.8927 | 1.1103 | 0.8471 |
| pets | arts & entertainment | 0.9369 | 1.1790 | 1.1361 | 1.0823 | 0.7266 | 0.3271 | 0.8947 | 0.9211 | 0.7920 | 0.5544 |
| pets | event planning & services | 1.0691 | 1.2750 | 1.1880 | 1.3965 | 1.0543 | 0.3677 | 0.9565 | 0.9612 | 1.1415 | 0.7731 |
| pets | home services | 1.5615 | 1.7300 | 1.6722 | 1.6607 | 1.3713 | 0.5748 | 1.2943 | 1.2846 | 1.4827 | 1.0487 |
| pets | hotels & travel | 1.1082 | 1.3098 | 1.2074 | 1.4217 | 0.9515 | 0.3809 | 1.0013 | 1.0124 | 1.2002 | 0.6958 |
| pets | local flavor | 0.8504 | 1.1893 | 1.1327 | 1.7307 | 0.7924 | 0.3150 | 0.8693 | 0.8546 | 1.5836 | 0.6068 |
| pets | local services | 1.2527 | 1.4258 | 1.4034 | 1.7608 | 1.2150 | 0.4267 | 0.9876 | 0.9861 | 1.6138 | 1.0097 |
| pets | nightlife | 1.0189 | 1.1281 | 1.0948 | 0.9407 | 0.8372 | 0.3498 | 0.8952 | 0.8504 | 0.6711 | 0.6511 |
| professional services | financial services | 0.9745 | 0.9269 | 0.9719 | 2.1008 | 0.7411 | 0.3194 | 0.4849 | 0.4889 | 2.0500 | 0.5196 |
| professional services | home services | 1.6797 | 1.7640 | 1.8521 | 1.5645 | 2.2404 | 0.6522 | 1.5261 | 1.5653 | 1.3707 | 1.3153 |
| professional services | religious organizations | nan | 0.8873 | 0.6747 | nan | nan | nan | 0.9792 | 1.1618 | nan | nan |
| public services & government | arts & entertainment | 0.9512 | 1.1804 | 1.1912 | 1.0547 | 0.9123 | 0.3373 | 0.9074 | 0.9191 | 0.7603 | 0.6983 |
| public services & government | education | 0.8932 | 1.0645 | 0.9430 | 1.9223 | 1.0481 | 0.3179 | 0.6890 | 0.6459 | 1.8739 | 0.8670 |

RMSE and MAE for domain-pairs in the Yelp dataset contd.

| source | target | CCA RMSE | CD-SVD RMSE | SD-SVD RMSE | RMGM RMSE | CMF RMSE | CCA MAE | CD-SVD MAE | SD-SVD MAE | RMGM MAE | CMF MAE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| public services & government | event planning & services | 1.0835 | 1.2594 | 1.3068 | 1.4268 | 1.0866 | 0.3757 | 0.9109 | 0.9542 | 1.1906 | 0.8351 |
| public services & government | financial services | 1.2015 | 1.0398 | 1.2670 | 1.7537 | 1.2650 | 0.4502 | 0.9116 | 0.9084 | 1.6658 | 1.1557 |
| public services & government | hotels & travel | 1.0867 | 1.1848 | 1.2052 | 1.4248 | 1.1364 | 0.3853 | 1.1436 | 1.0917 | 1.2197 | 0.8611 |
| public services & government | local flavor | 0.9090 | 1.1286 | 1.2048 | 1.6823 | 0.8675 | 0.3230 | 0.8247 | 0.8795 | 1.5260 | 0.6772 |
| public services & government | mass media | nan | 1.5091 | 1.5893 | nan | nan | nan | 1.1619 | 1.2104 | nan | nan |
| religious organizations | education | 1.3924 | 1.6810 | 1.5882 | 1.7111 | 5.0885 | 0.5074 | 1.1963 | 1.4445 | 1.6025 | 3.3215 |
| religious organizations | professional services | 1.4449 | 0.8288 | 0.6279 | 1.8179 | 1.4915 | 0.6115 | 0.8975 | 0.9419 | 1.7590 | 1.4793 |
| restaurants | food | 1.0543 | 1.1137 | 1.1616 | 1.2997 | 1.0723 | 0.3625 | 0.8636 | 0.8846 | 1.0073 | 0.7643 |
| restaurants | nightlife | 1.0697 | 1.1484 | 1.1733 | 1.2975 | 1.8423 | 0.3748 | 0.8990 | 0.9188 | 1.0301 | 0.7662 |
| shopping | arts & entertainment | 0.9643 | 1.0702 | 1.1035 | 1.3083 | 1.2578 | 0.3343 | 0.8422 | 0.8427 | 1.0333 | 0.7224 |
| shopping | food | 1.0369 | 1.1108 | 1.1405 | 1.0342 | 0.8487 | 0.3577 | 0.8667 | 0.8917 | 0.7397 | 0.6453 |
| shopping | nightlife | 1.0084 | 1.1100 | 1.1122 | 0.9749 | 1.0953 | 0.3506 | 0.8724 | 0.8920 | 0.7002 | 0.7376 |

## B.3 COLD-START ANALYSIS OF DOMAIN PAIRS FOR YELP DATASET

### B.3.1 MAEs for Target User Profiles

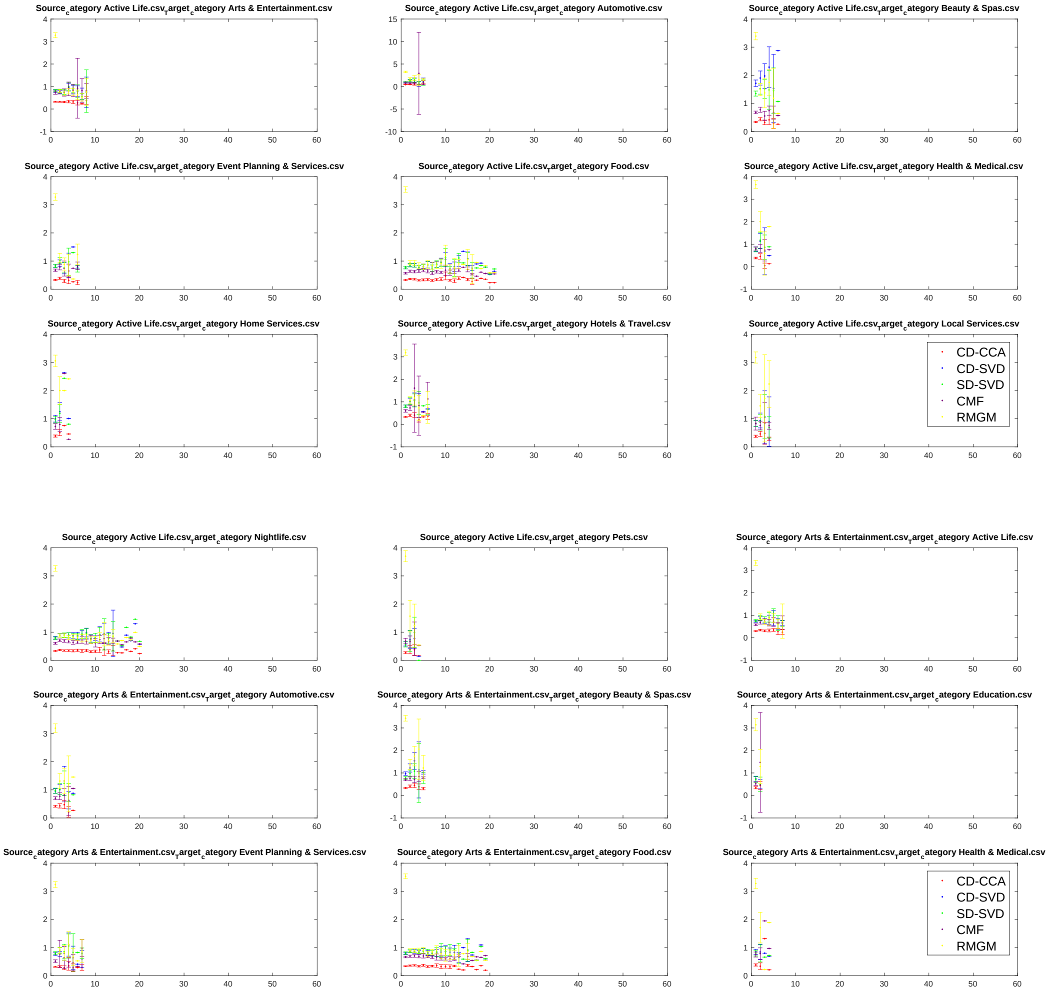


Figure 74: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' target domain profile size

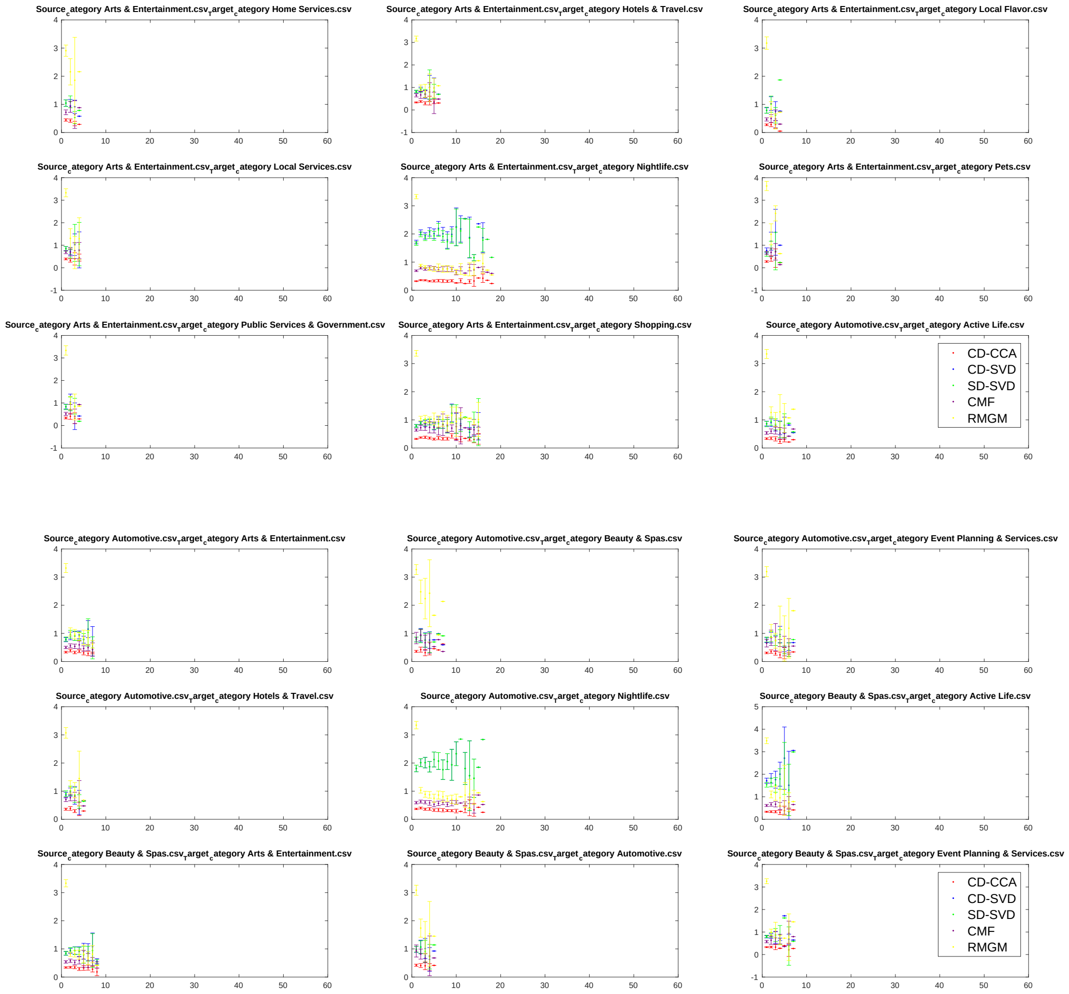


Figure 74: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 19 to 36

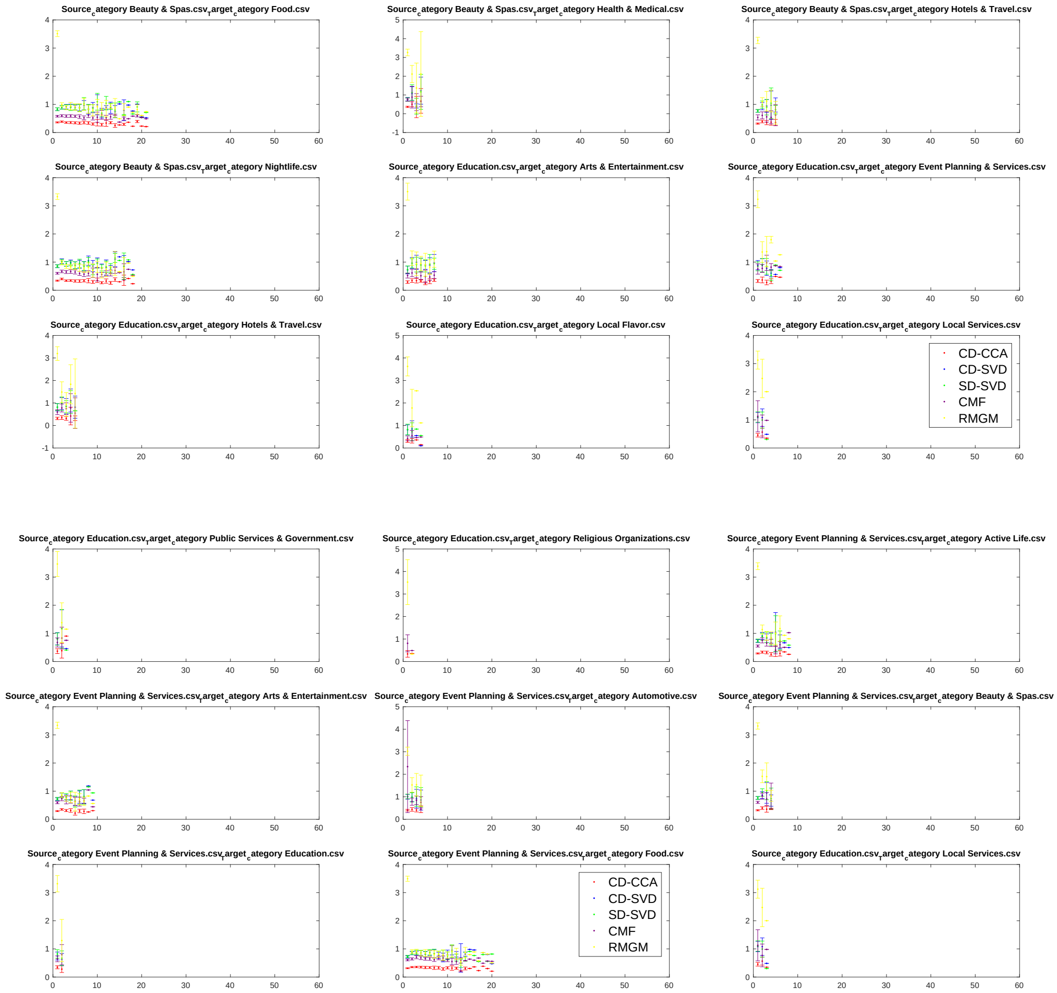


Figure 74: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 37 to 53

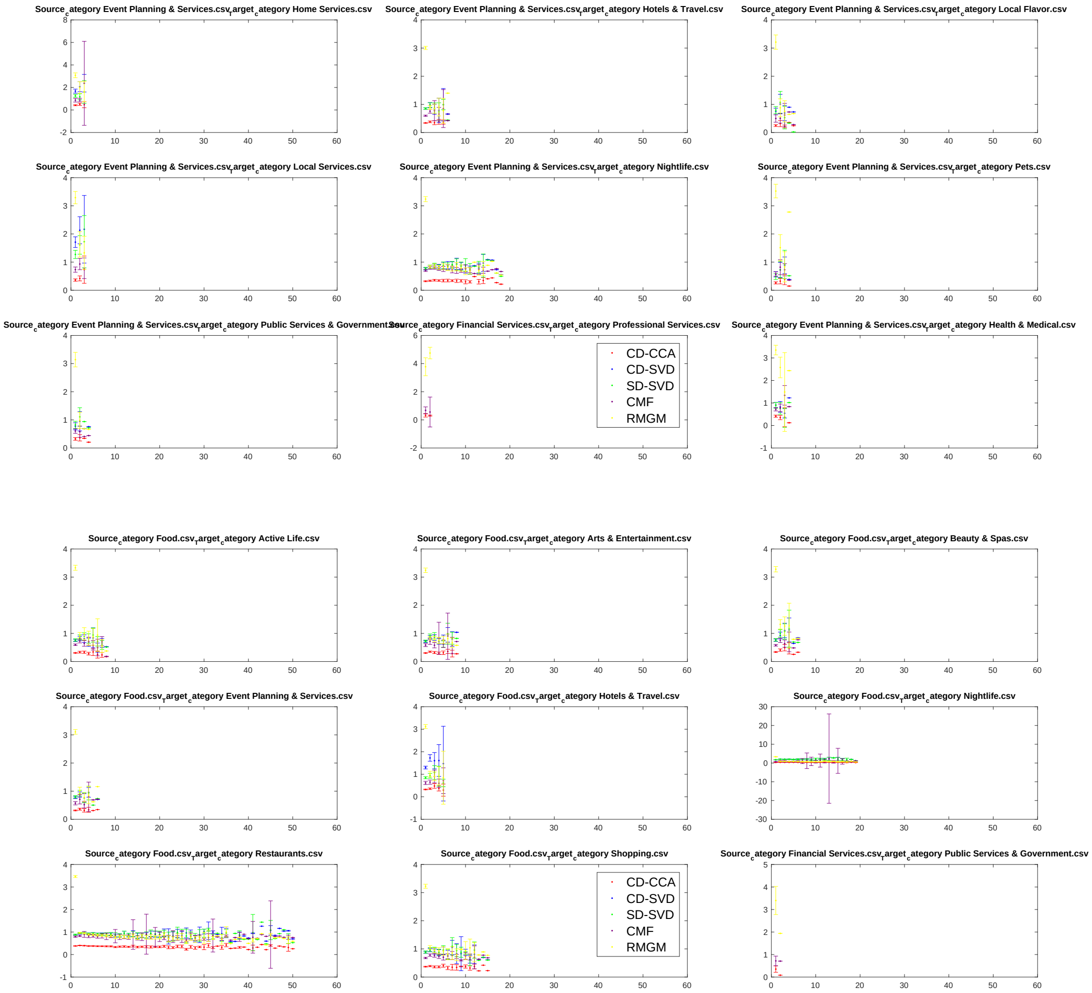


Figure 74: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 54 to 71

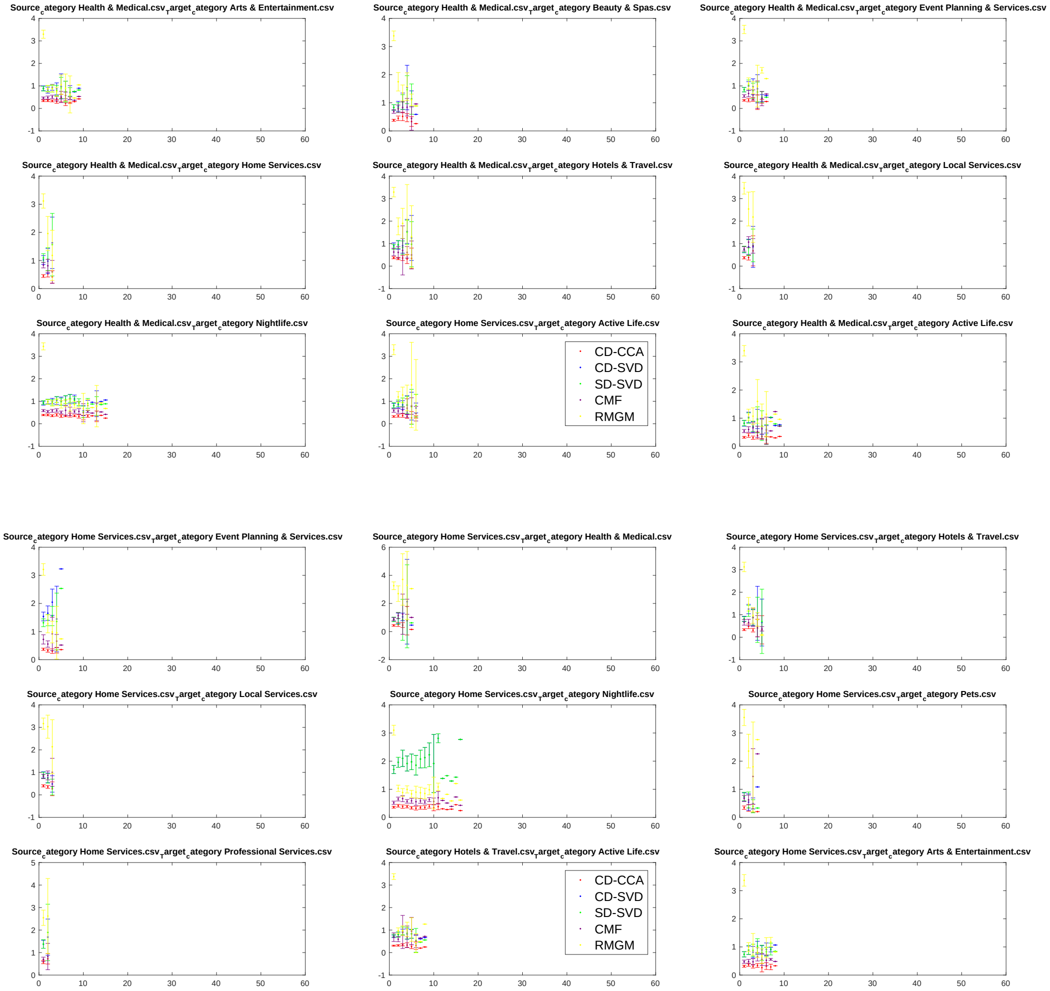


Figure 74: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 72 to 89

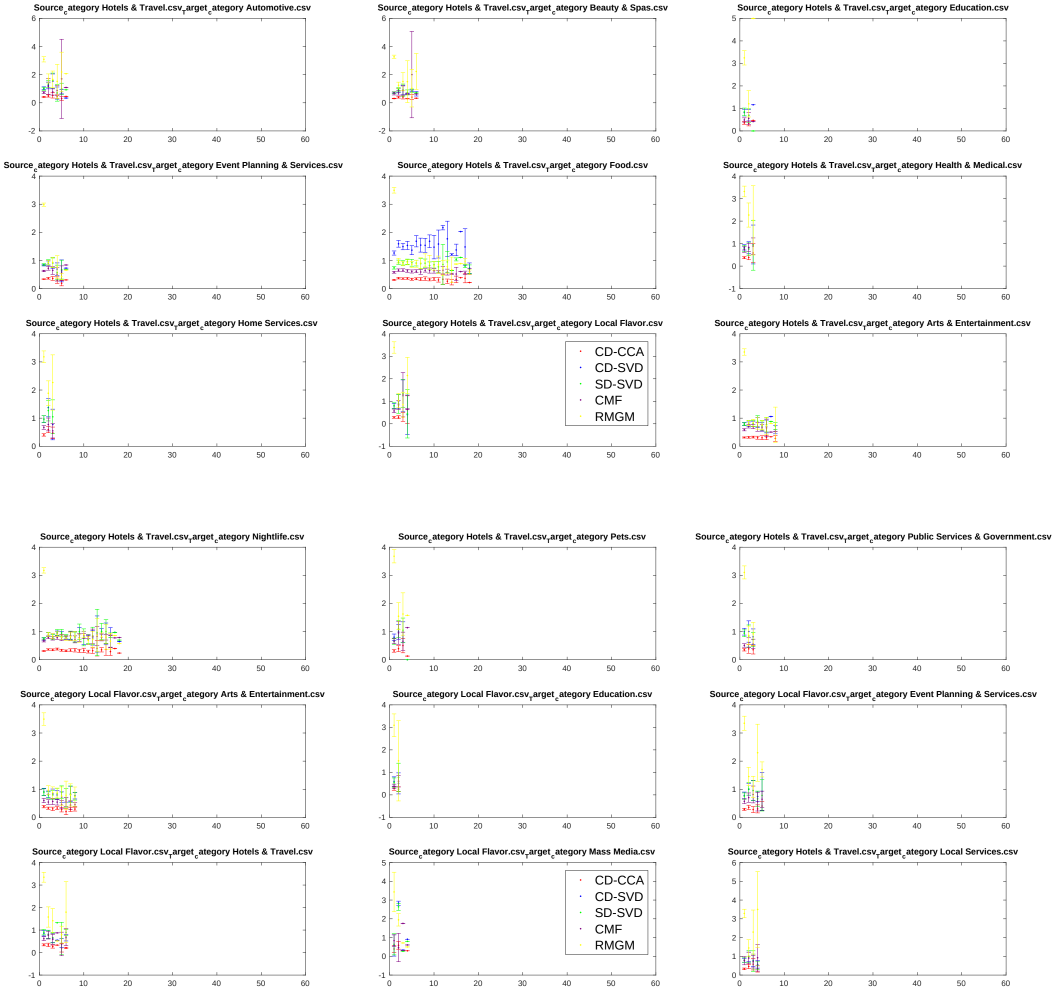


Figure 74: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 90 to 107

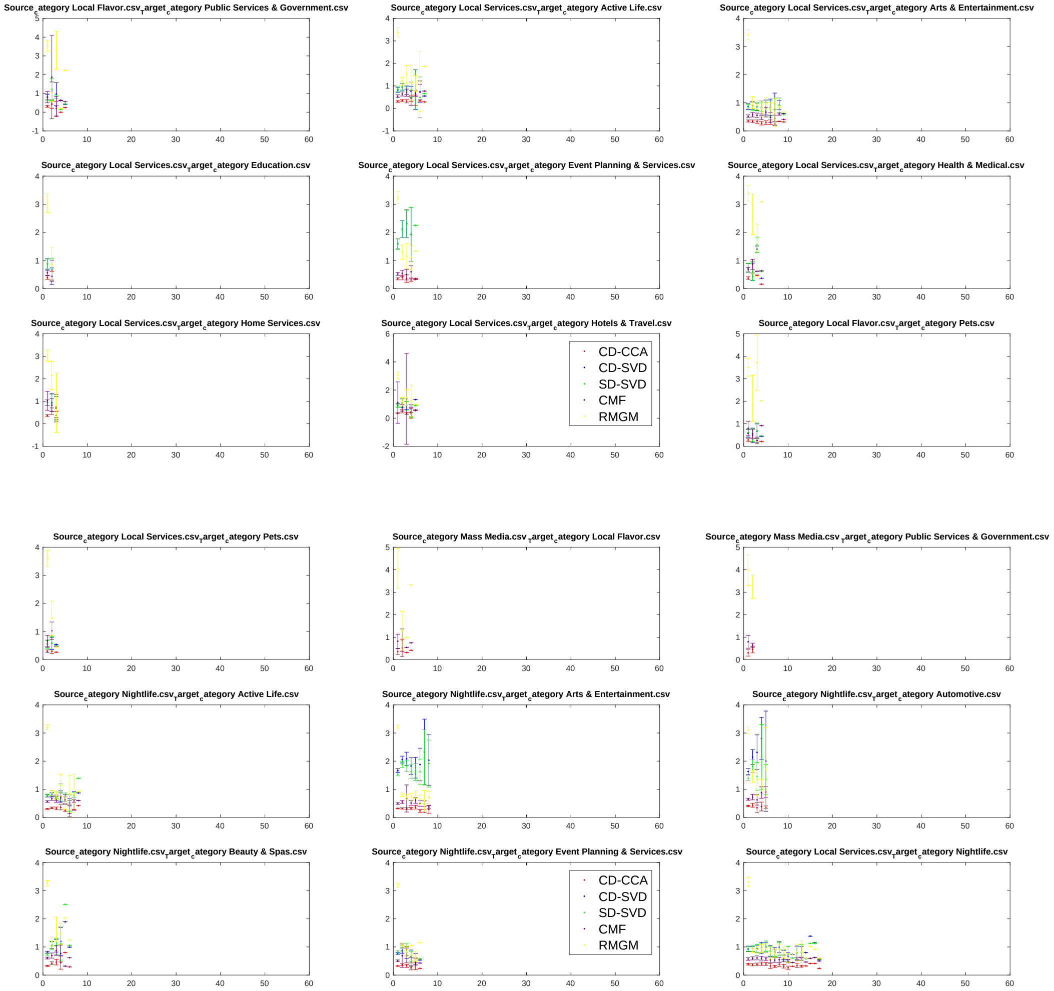


Figure 74: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 108 to 125

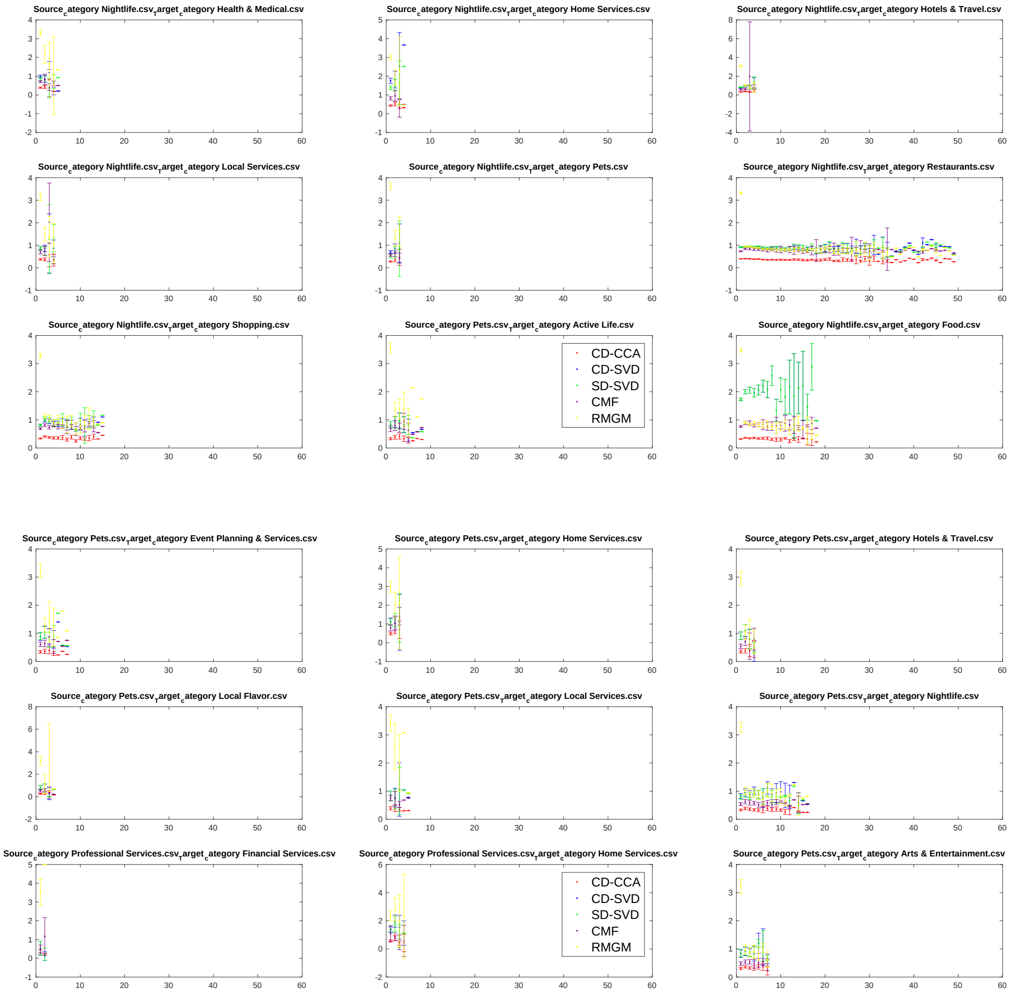


Figure 74: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 126 to 143

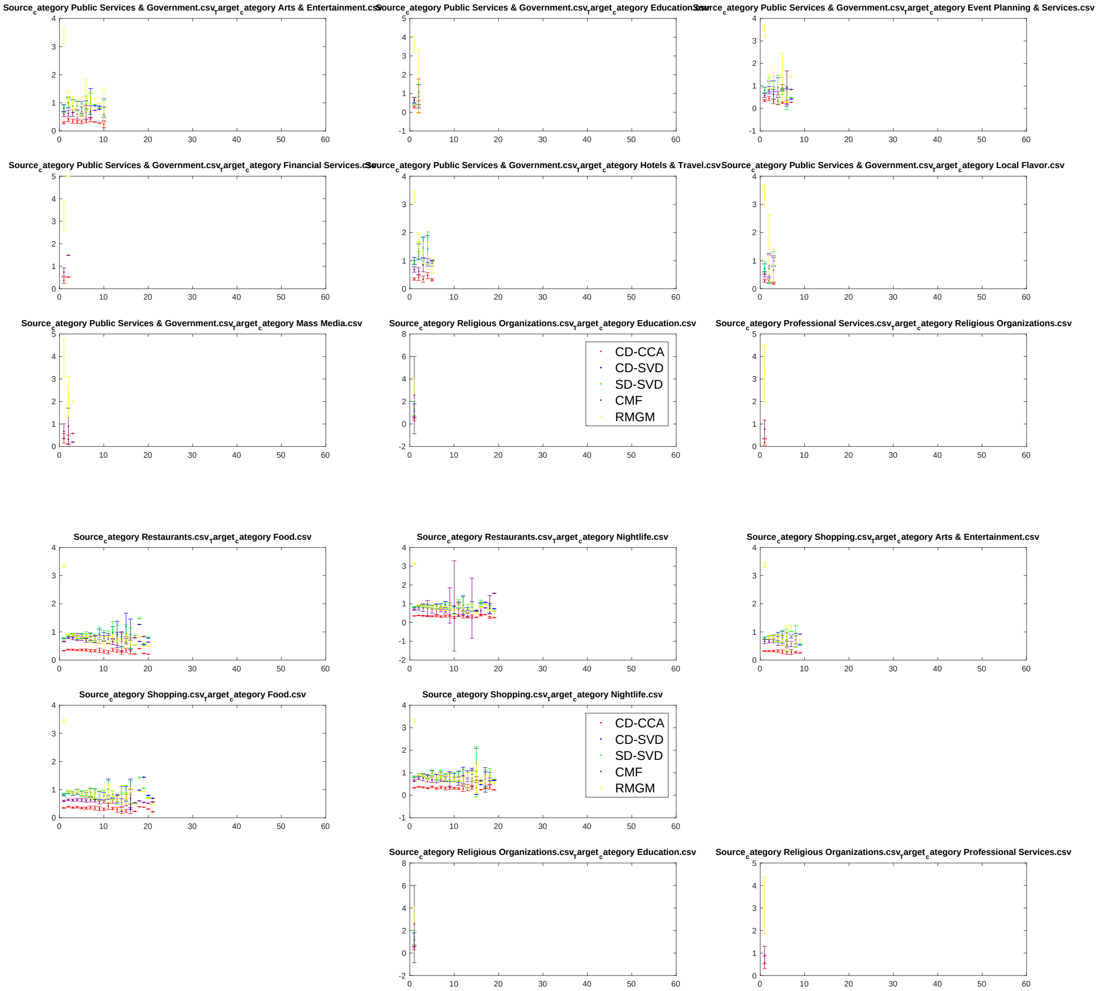


Figure 74: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 144 to 158

### B.3.2 MAEs for Source User Profiles

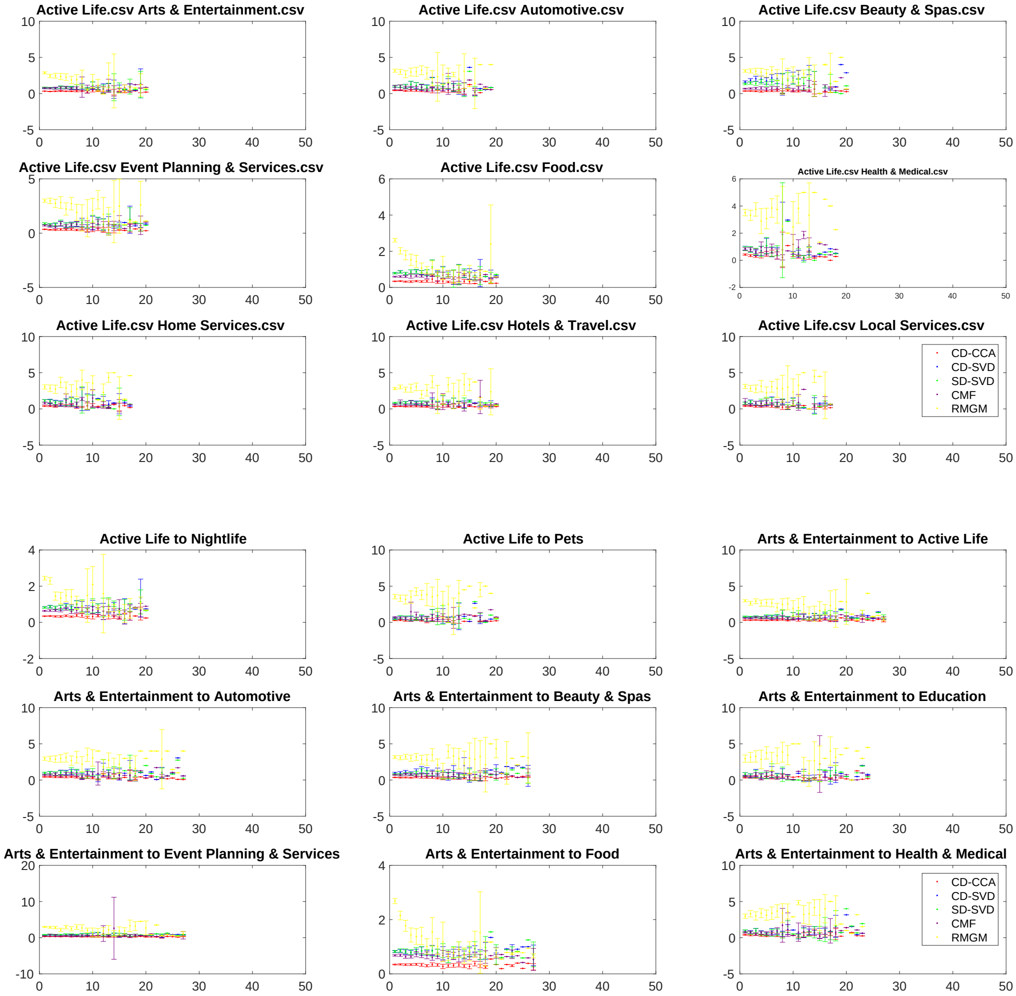


Figure 75: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' source domain profile size

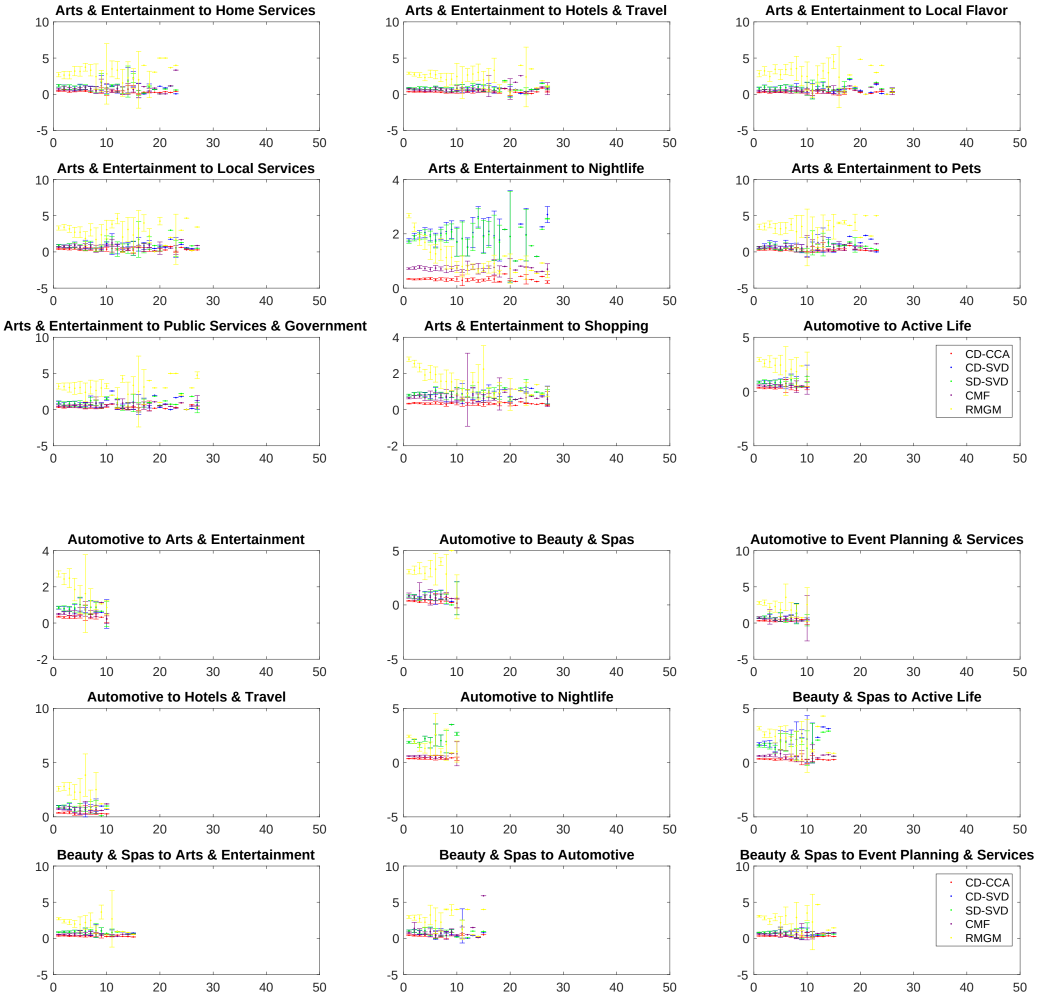


Figure 75: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' source domain profile size for domain pairs 19 to 36

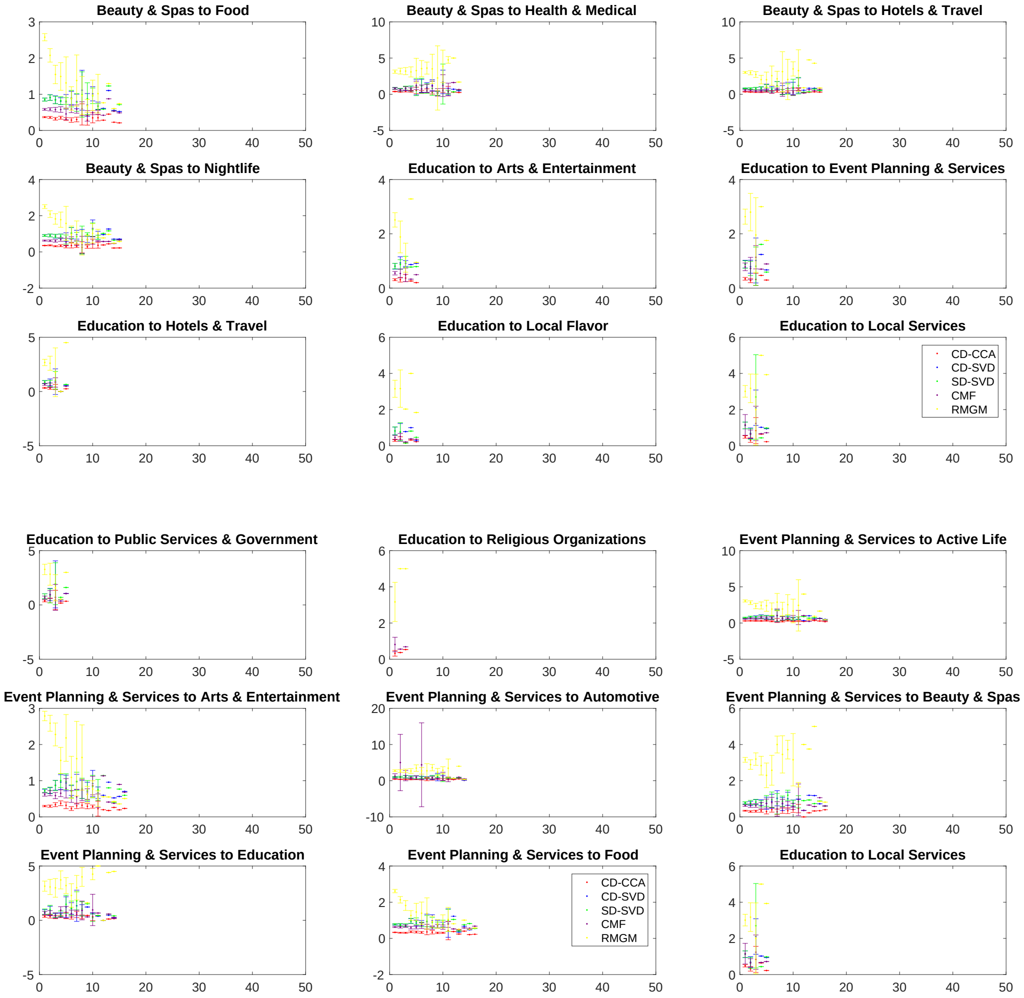


Figure 75: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' source domain profile size for domain pairs 37 to 53

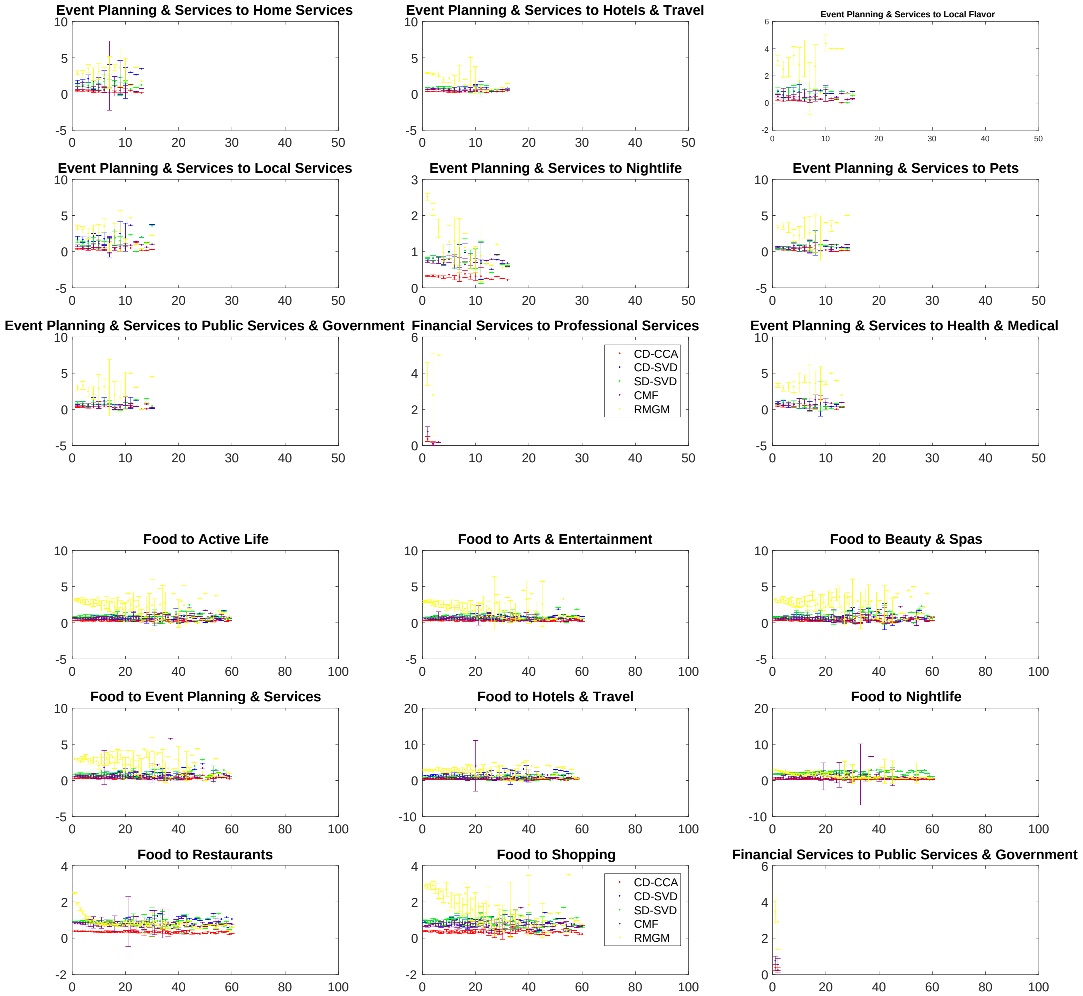


Figure 75: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' source domain profile size for domain pairs 54 to 71

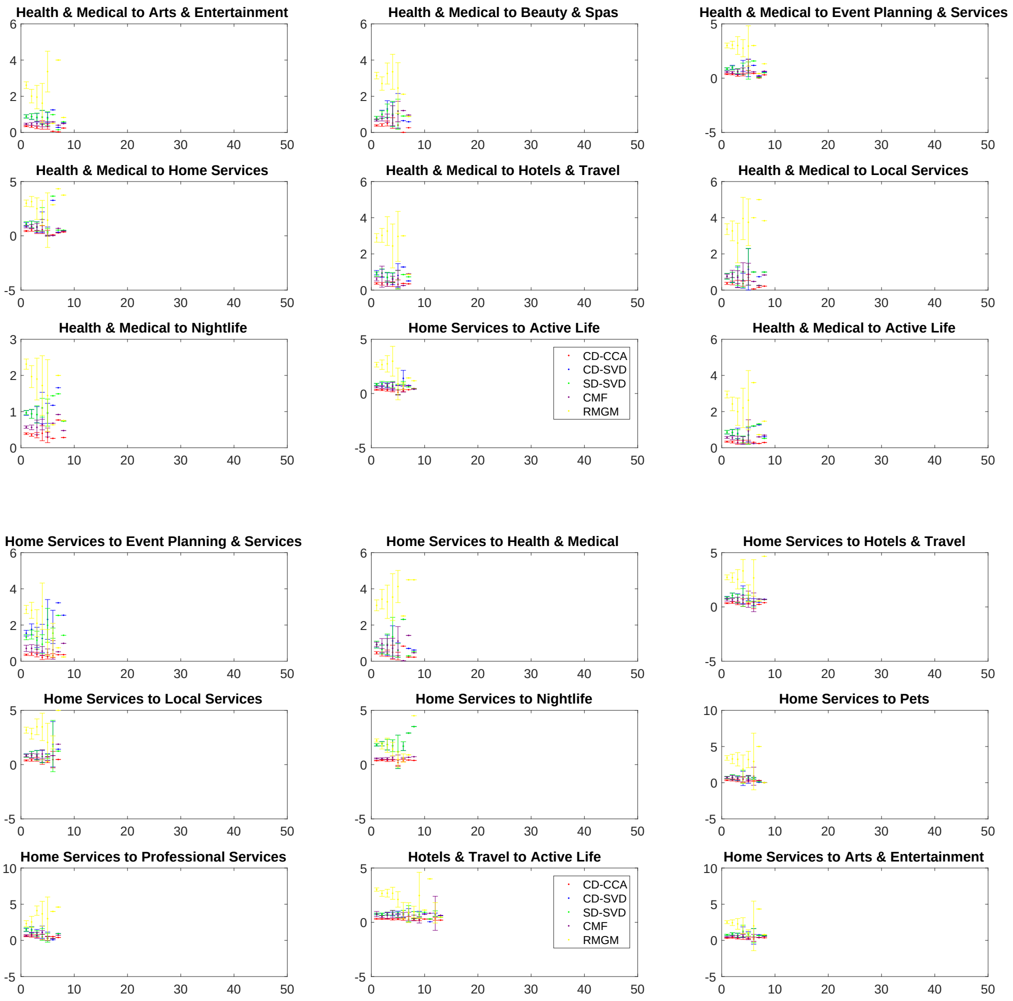


Figure 75: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' source domain profile size for domain pairs 72 to 89

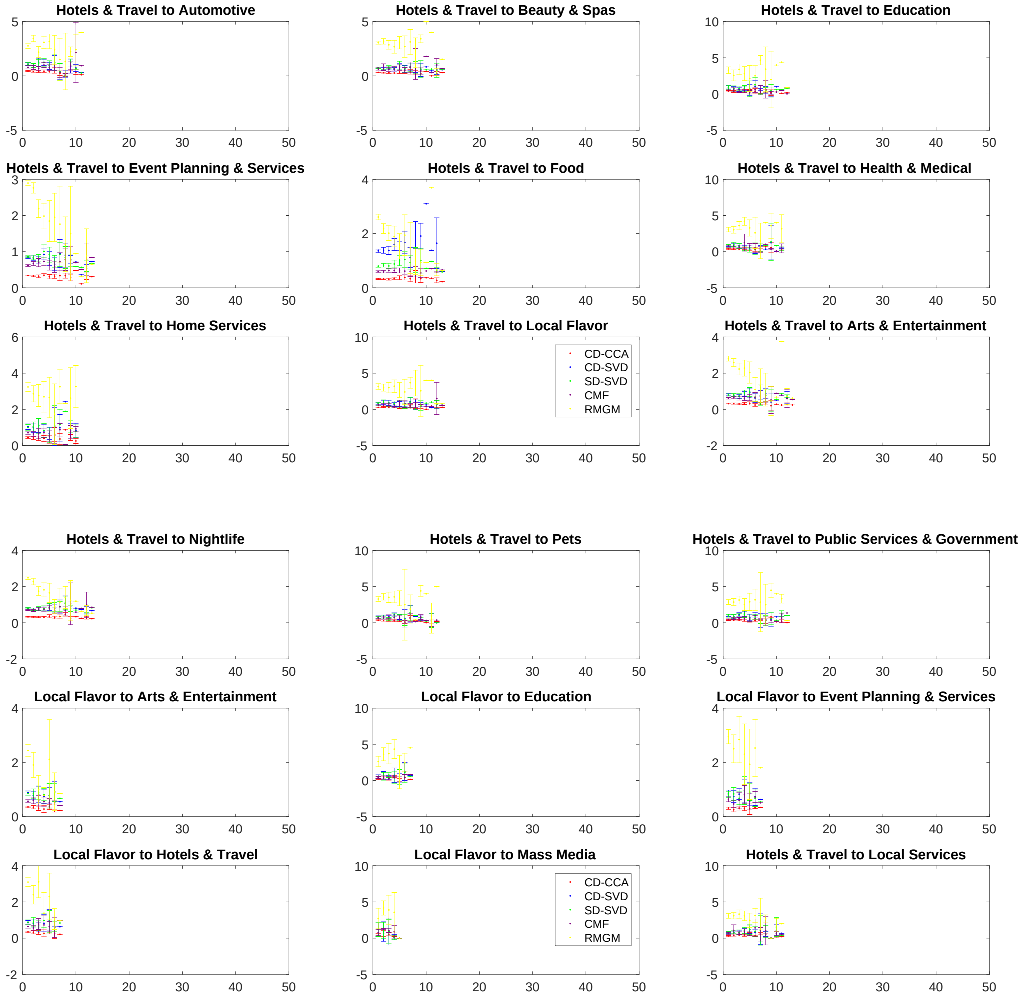


Figure 75: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size for domain pairs 90 to 107

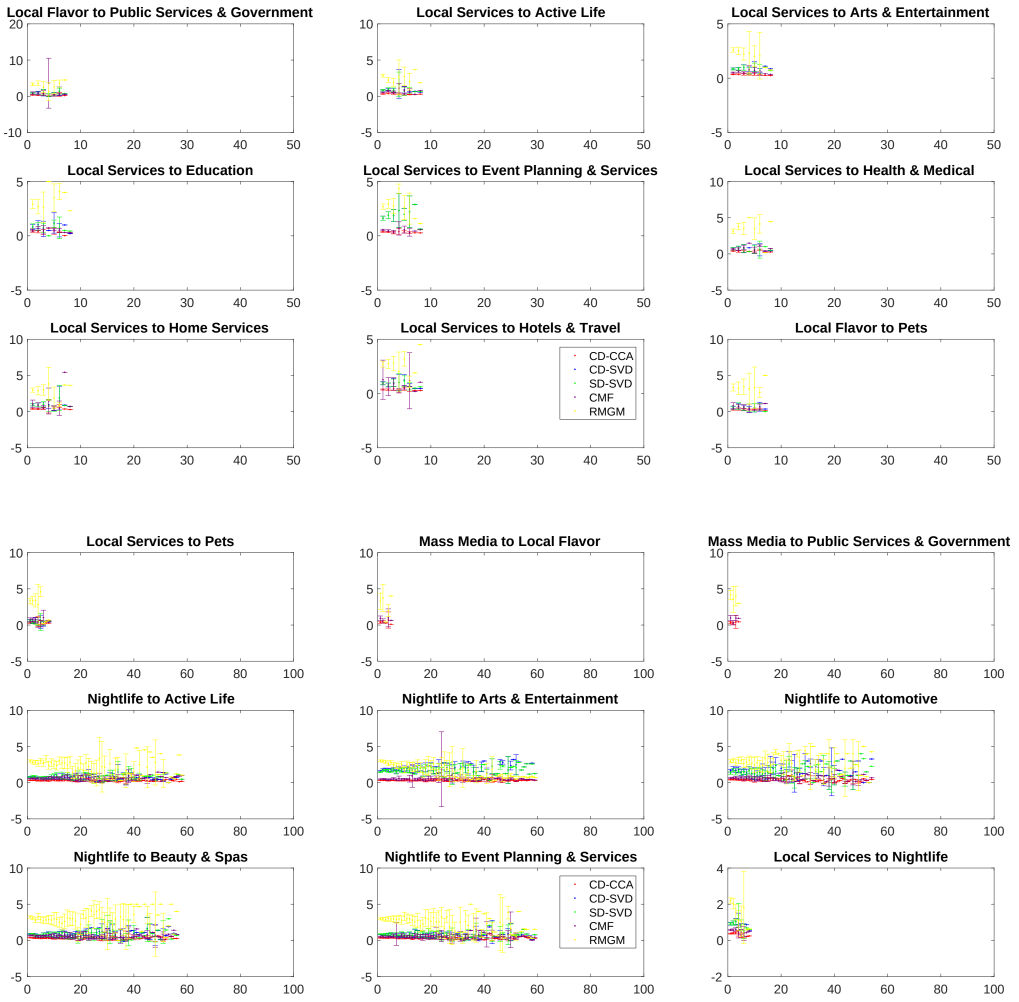


Figure 75: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size for domain pairs 108 to 125

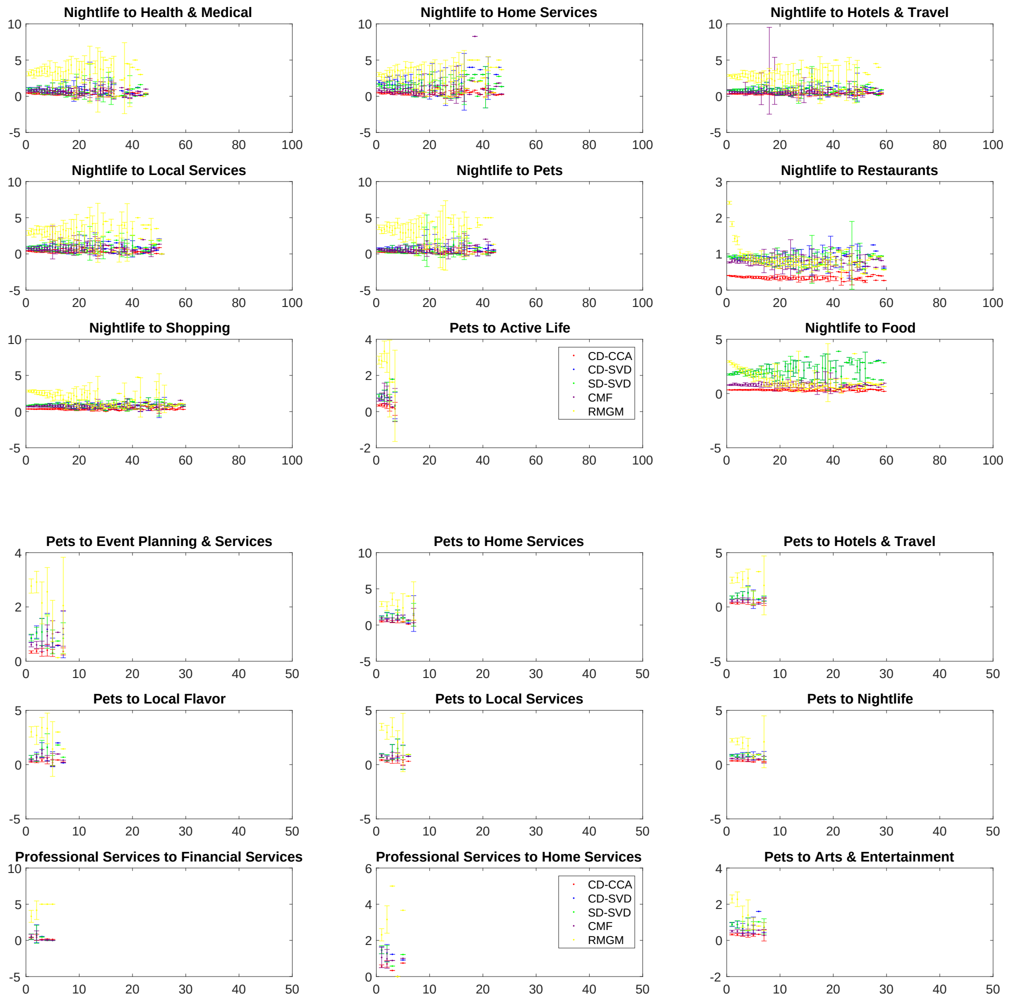


Figure 75: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size for domain pairs 126 to 143

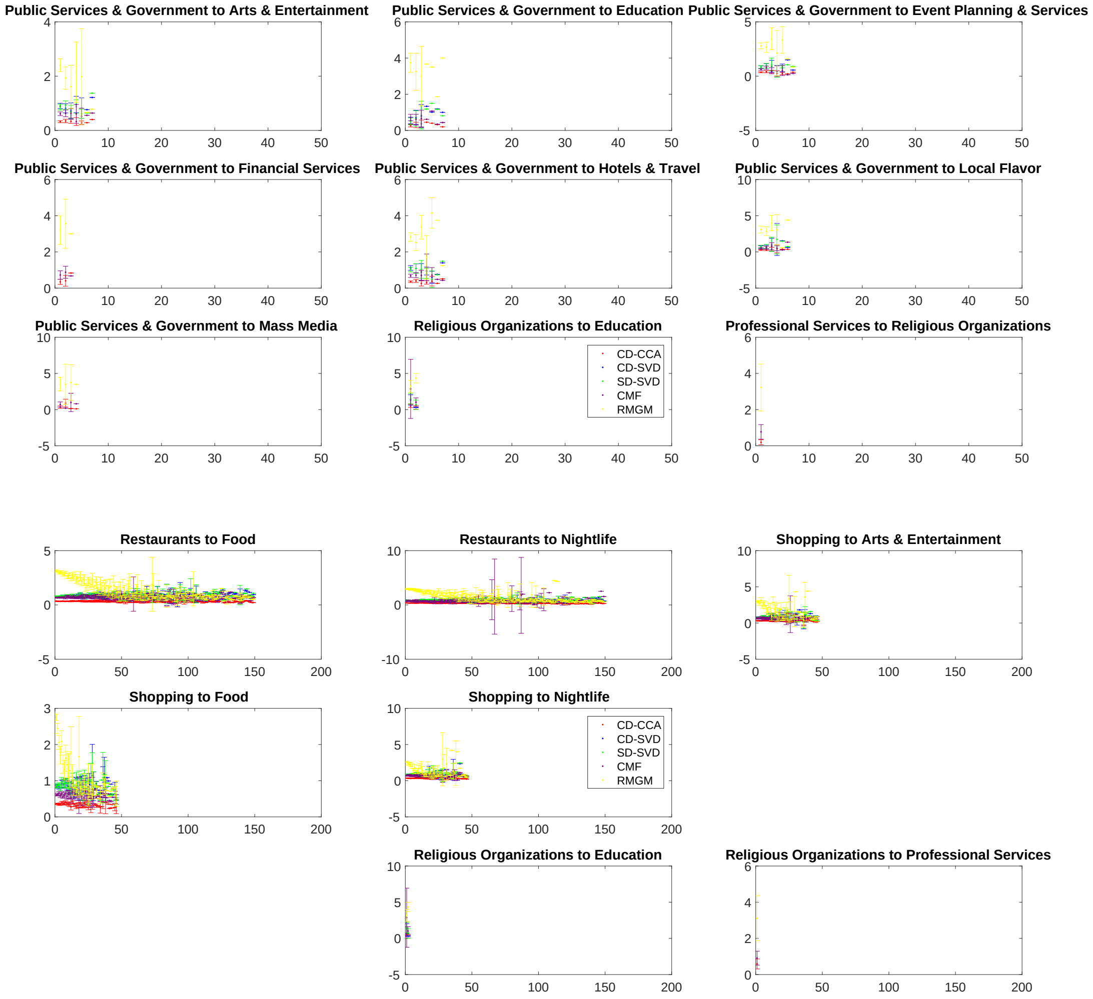


Figure 75: User-based MAE of algorithms in the Yelp dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size for domain pairs 144 to 158

## B.4 CORRELATION ANALYSIS FOR YELP DATASET

### B.4.1 Correlation Analysis Plots for the RMSE of Algorithms in the Yelp Dataset

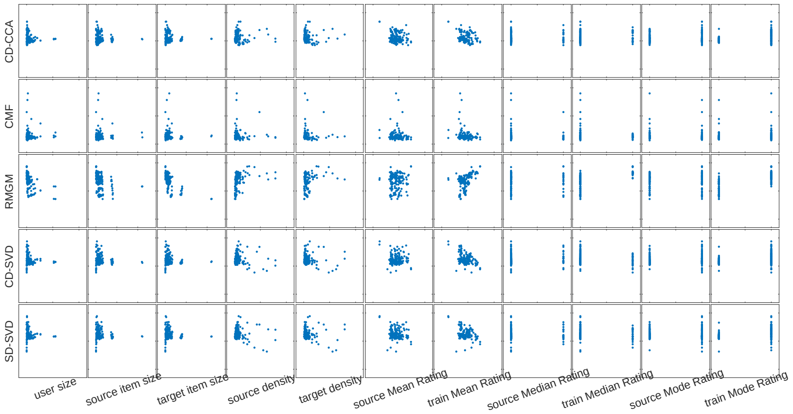


Figure 76: Scatter plot of general and central tendency statistics with the RMSE of algorithms in the Yelp dataset

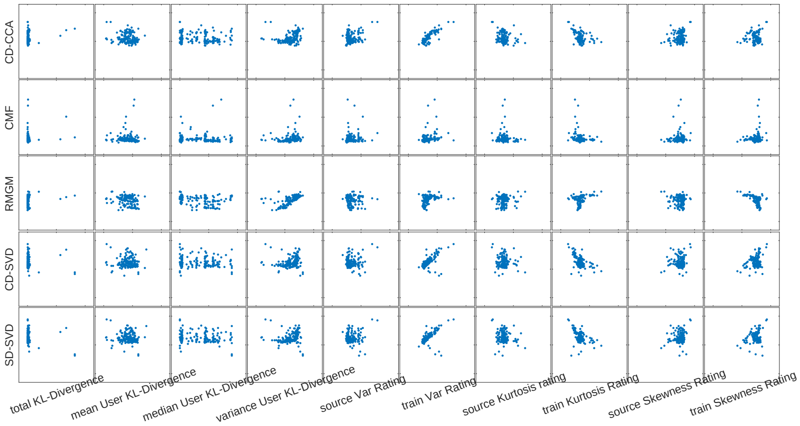


Figure 77: Scatter plot of dispersion statistics with the RMSE of algorithms in the Yelp dataset

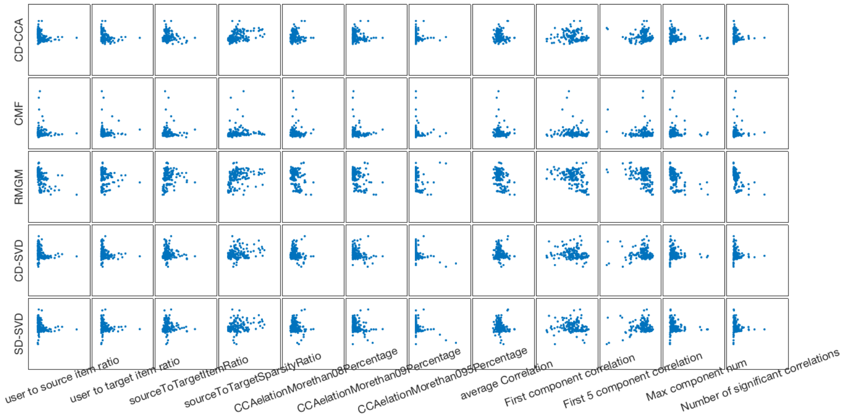


Figure 78: Scatter plot of CCA-related statistics with the RMSE of algorithms in the Yelp dataset

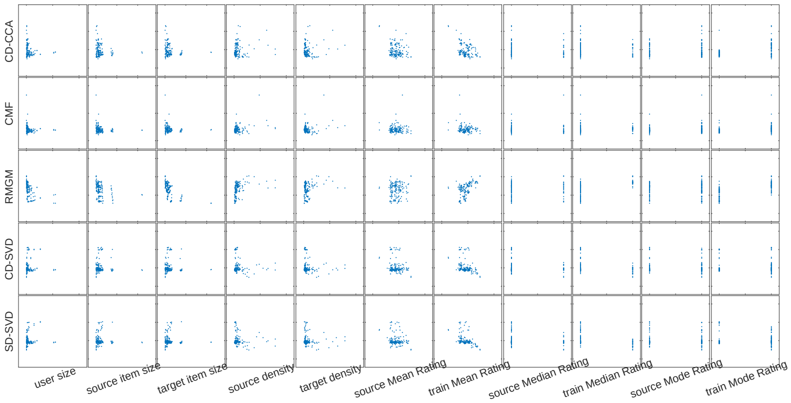


Figure 79: Scatter plot of general and central tendency statistics with the MAE of algorithms in the Yelp dataset

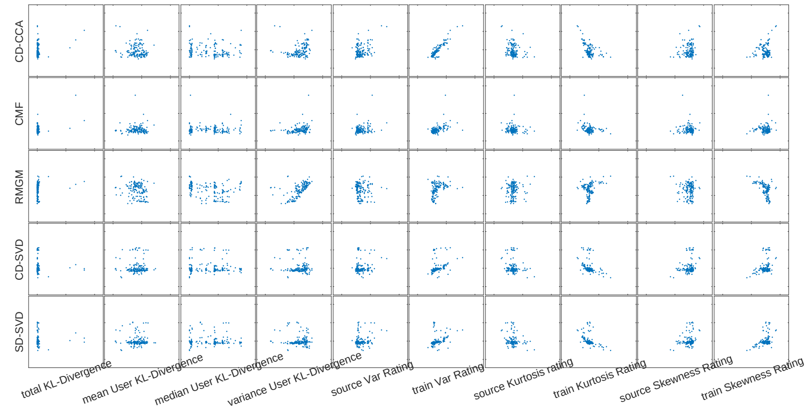


Figure 80: Scatter plot of dispersion statistics with the MAE of algorithms in the Yelp dataset

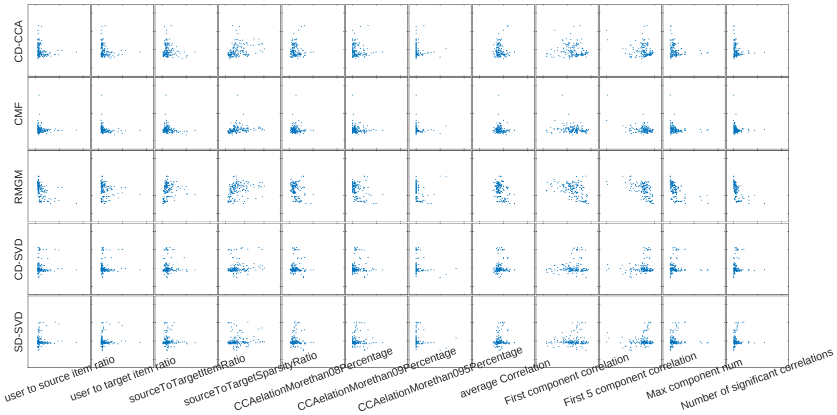


Figure 81: Scatter plot of CCA-related statistics with the MAE of algorithms in the Yelp dataset

### B.4.2 Correlation Analysis Plots for the Improvement Ratio of Algorithms in the Yelp Dataset

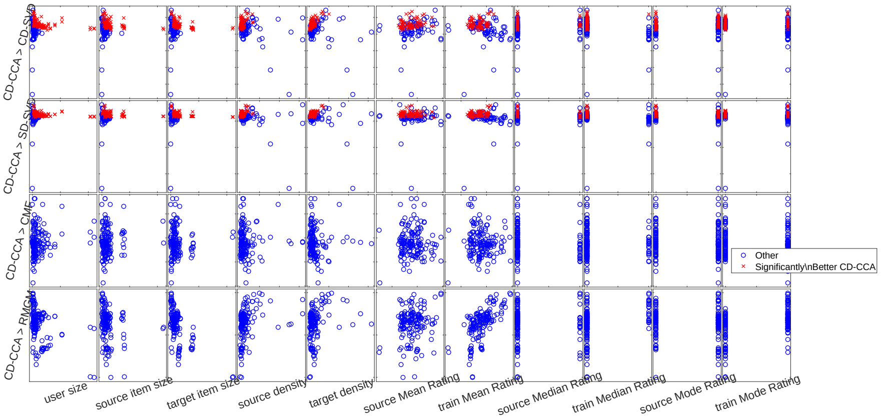


Figure 82: Scatter plot of general and central tendency statistics with the improvement ratio of CD-CCA over other algorithms in the Yelp dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms

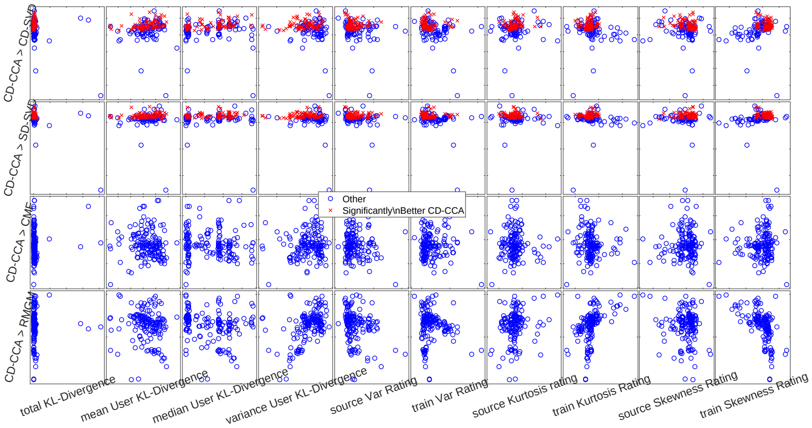


Figure 83: Scatter plot of dispersion statistics with the improvement ratio of CD-CCA over other algorithms in the Yelp dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms

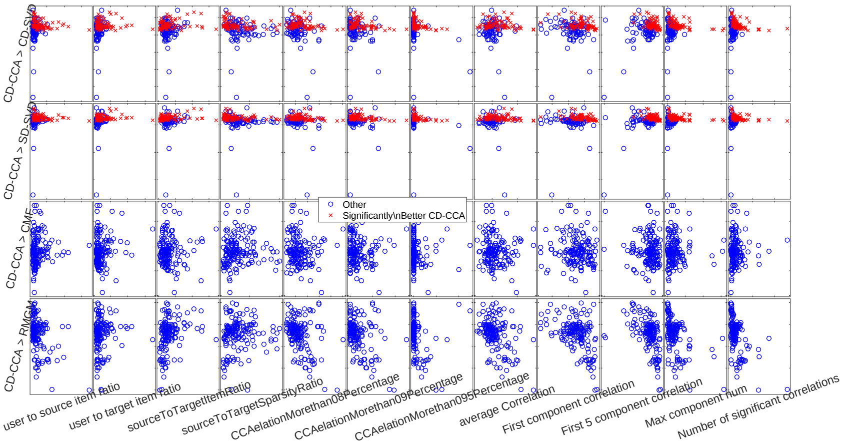


Figure 84: Scatter plot of CCA-related statistics with the improvement ratio of CD-CCA over other algorithms in the Yelp dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms

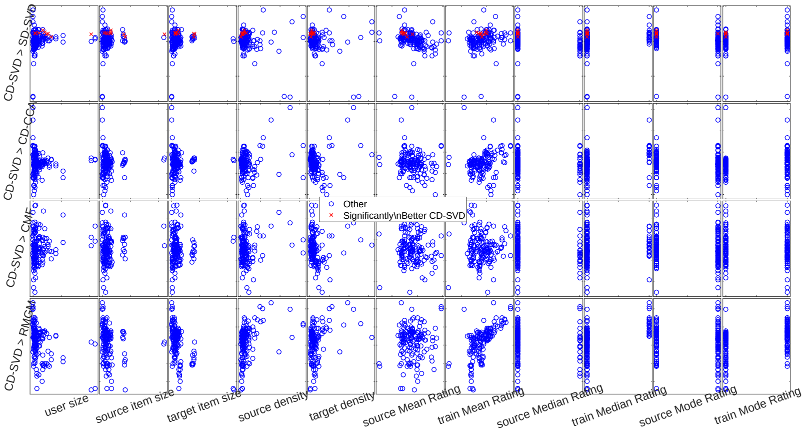


Figure 85: Scatter plot of general and central tendency statistics with the improvement ratio of CD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms

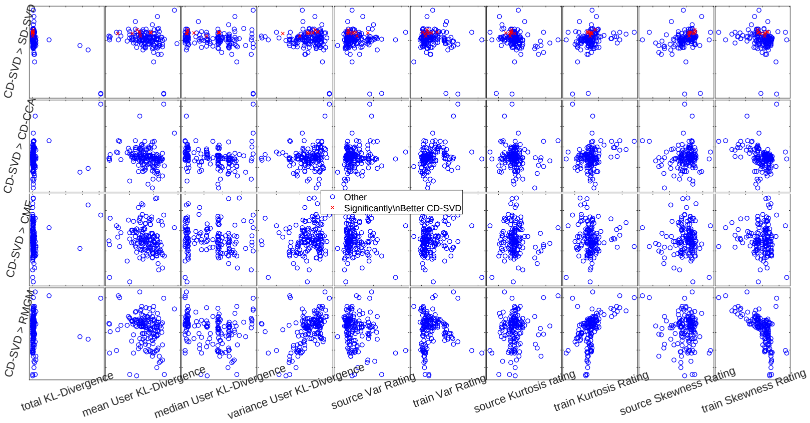


Figure 86: Scatter plot of dispersion statistics with the improvement ratio of CD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms

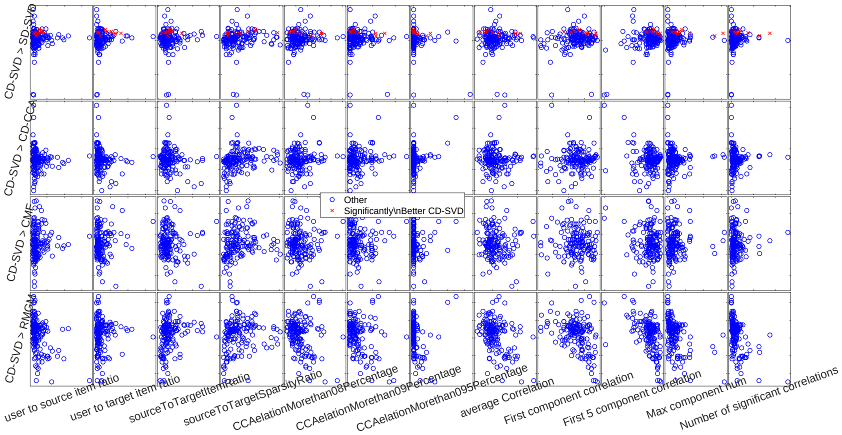


Figure 87: Scatter plot of CCA-related statistics with the improvement ratio of CD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms

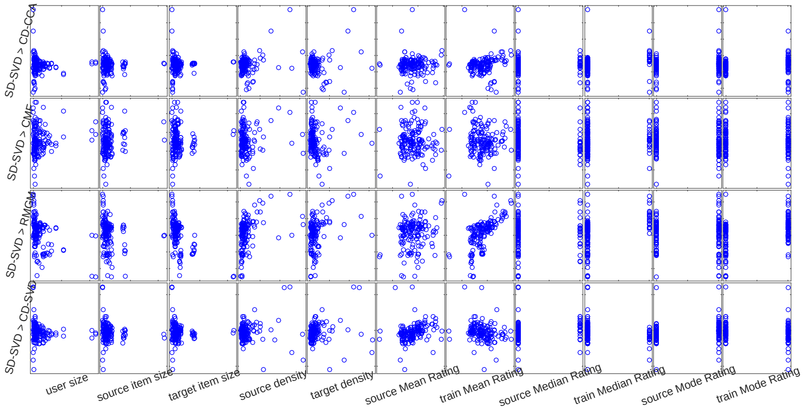


Figure 88: Scatter plot of general and central tendency statistics with the improvement ratio of SD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms

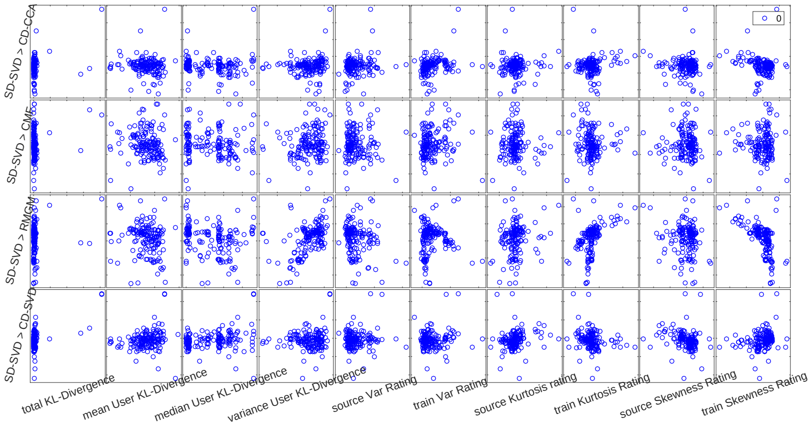


Figure 89: Scatter plot of dispersion statistics with the improvement ratio of SD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms

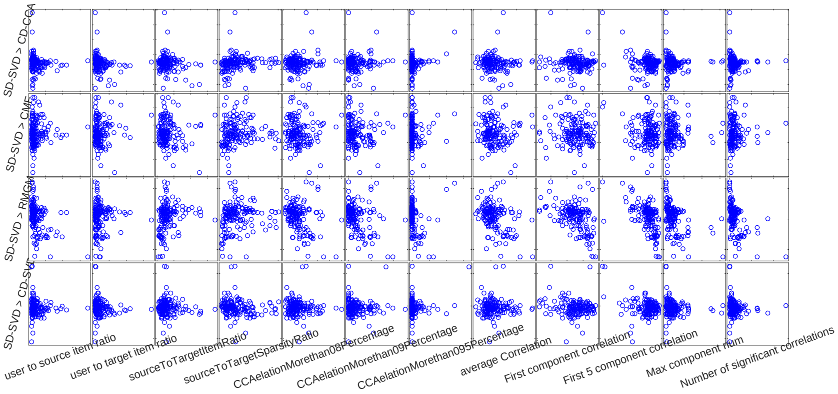


Figure 90: Scatter plot of CCA-related statistics with the improvement ratio of SD-SVD over other algorithms in the Yelp dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms

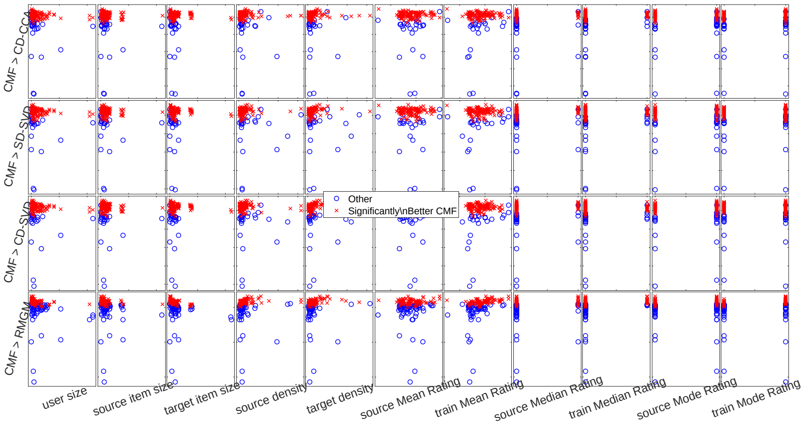


Figure 91: Scatter plot of general and central tendency statistics with the improvement ratio of CMF over other algorithms in the Yelp dataset; red cross shows the cases in which CMF is significantly better than other algorithms

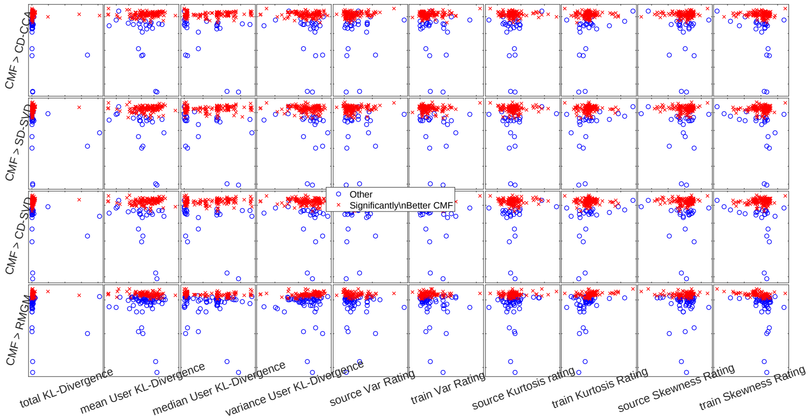


Figure 92: Scatter plot of dispersion statistics with the improvement ratio of CMF over other algorithms in the Yelp dataset; red cross shows the cases in which CMF is significantly better than other algorithms

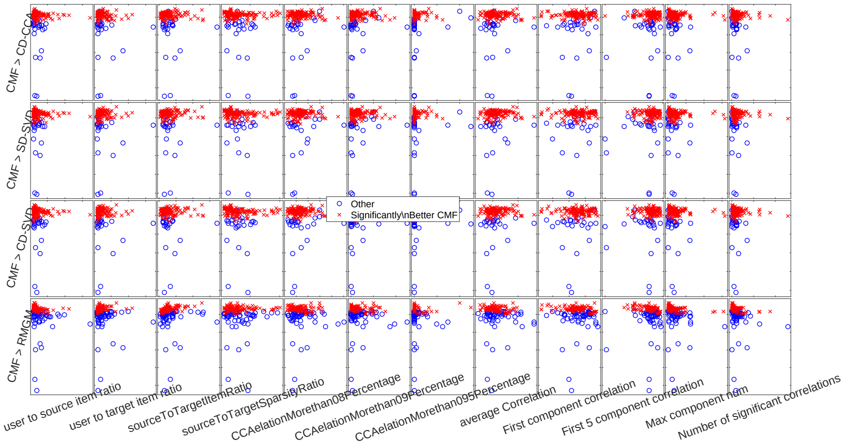


Figure 93: Scatter plot of CCA-related statistics with the improvement ratio of CMF over other algorithms in the Yelp dataset; red cross shows the cases in which CMF is significantly better than other algorithms

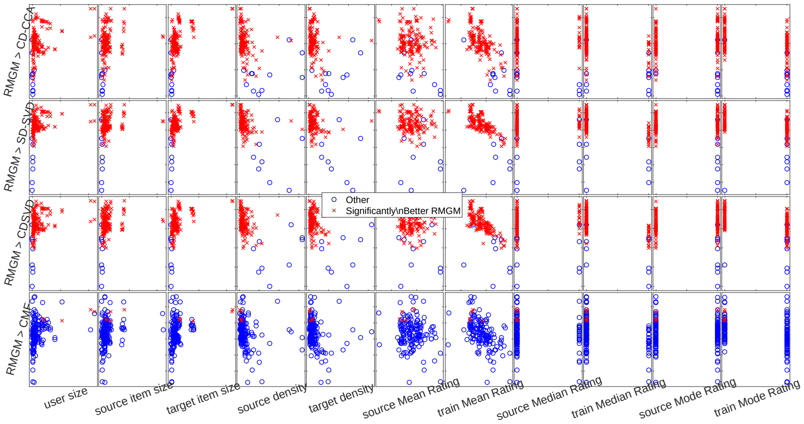


Figure 94: Scatter plot of general and central tendency statistics with the improvement ratio of RMGM over other algorithms in the Yelp dataset; red cross shows the cases in which RMGM is significantly better than other algorithms

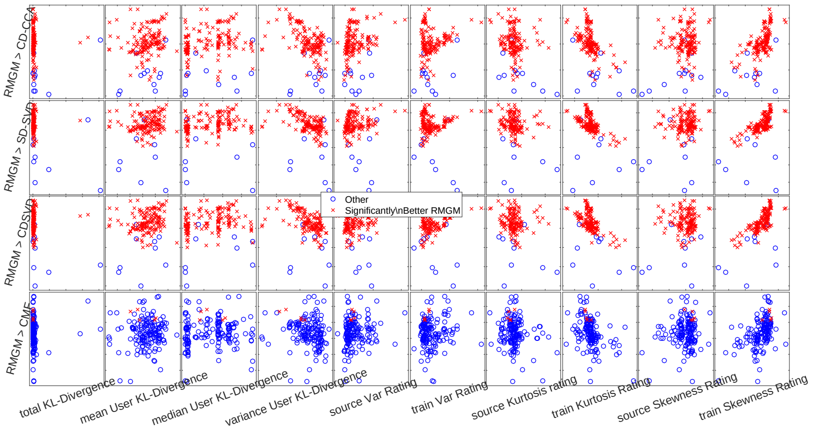


Figure 95: Scatter plot of dispersion statistics with the improvement ratio of RMGM over other algorithms in the Yelp dataset; red cross shows the cases in which RMGM is significantly better than other algorithms

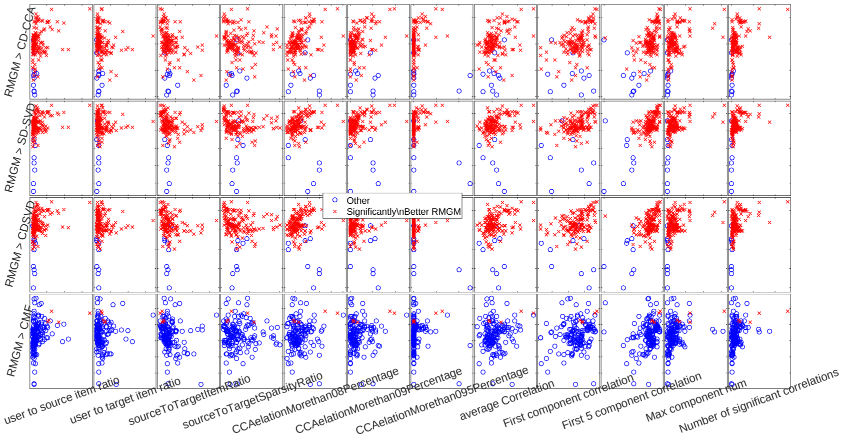


Figure 96: Scatter plot of CCA-related statistics with the improvement ratio of RMGM over other algorithms in the Yelp dataset; red cross shows the cases in which RMGM is significantly better than other algorithms

# APPENDIX C

# IMHONET DATA FIGURES AND TABLES

## C.1 DOMAIN PAIR STATISTICS FOR IMHONET DATASET

Table 48: Domain and domain-pair data size statistics for the Imhonet dataset

| source | target | user size | source item size | target item size | source density | target density | user to source item ratio | user to target item ratio | source to target item ratio | source to target density ratio |
|---|---|---|---|---|---|---|---|---|---|---|
| book | game | 41756 | 125688 | 11407 | 0.0007 | 0.0020 | 0.3322 | 3.6606 | 11.0185 | 0.3574 |
| book | movie | 186877 | 155765 | 85892 | 0.0003 | 0.0014 | 1.1997 | 2.1757 | 1.8135 | 0.2501 |
| book | perfume | 16750 | 105805 | 3545 | 0.0011 | 0.0037 | 0.1583 | 4.7250 | 29.8463 | 0.2836 |
| game | book | 41756 | 11407 | 125688 | 0.0020 | 0.0007 | 3.6606 | 0.3322 | 0.0908 | 2.7977 |
| game | movie | 49784 | 11715 | 75599 | 0.0019 | 0.0028 | 4.2496 | 0.6585 | 0.1550 | 0.6754 |
| game | perfume | 6297 | 6854 | 3232 | 0.0030 | 0.0041 | 0.9187 | 1.9483 | 2.1207 | 0.7233 |
| movie | book | 186877 | 85892 | 155765 | 0.0014 | 0.0003 | 2.1757 | 1.1997 | 0.5514 | 3.9989 |
| movie | game | 49784 | 75599 | 11715 | 0.0028 | 0.0019 | 0.6585 | 4.2496 | 6.4532 | 1.4806 |
| movie | perfume | 17882 | 63708 | 3565 | 0.0041 | 0.0037 | 0.2807 | 5.0160 | 17.8704 | 1.1213 |
| perfume | book | 16750 | 3545 | 105805 | 0.0037 | 0.0011 | 4.7250 | 0.1583 | 0.0335 | 3.5262 |
| perfume | game | 6297 | 3232 | 6854 | 0.0041 | 0.0030 | 1.9483 | 0.9187 | 0.4715 | 1.3826 |
| perfume | movie | 17882 | 3565 | 63708 | 0.0037 | 0.0041 | 5.0160 | 0.2807 | 0.0560 | 0.8918 |

Table 49: Domain ratings central tendency and dispersion statistics for the Imhonet dataset

| source | target | source mean rating | target mean rating | source median rating | target median rating | source mode rating | target mode rating | source var. rating | target var. rating |
|---|---|---|---|---|---|---|---|---|---|
| book | game | 7.7312 | 7.5362 | 8 | 8 | 8 | 8 | 4.1508 | 5.0477 |
| book | movie | 7.6869 | 7.2340 | 8 | 8 | 8 | 8 | 4.6493 | 5.0026 |
| book | perfume | 7.7453 | 7.1408 | 8 | 8 | 8 | 8 | 4.1754 | 5.3937 |
| game | book | 7.5362 | 7.7312 | 8 | 8 | 8 | 8 | 5.0477 | 4.1508 |
| game | movie | 7.5126 | 7.2509 | 8 | 8 | 10 | 8 | 5.2903 | 4.5204 |
| game | perfume | 7.5926 | 7.0892 | 8 | 8 | 8 | 8 | 4.4979 | 5.6549 |
| movie | book | 7.2340 | 7.6869 | 8 | 8 | 8 | 8 | 5.0026 | 4.6493 |
| movie | game | 7.2509 | 7.5126 | 8 | 8 | 8 | 10 | 4.5204 | 5.2903 |
| movie | perfume | 7.2722 | 7.1548 | 8 | 8 | 8 | 8 | 4.4203 | 5.4384 |
| perfume | book | 7.1408 | 7.7453 | 8 | 8 | 8 | 8 | 5.3937 | 4.1754 |
| perfume | game | 7.0892 | 7.5926 | 8 | 8 | 8 | 8 | 5.6549 | 4.4979 |
| perfume | movie | 7.1548 | 7.2722 | 8 | 8 | 8 | 8 | 5.4384 | 4.4203 |

Table 50: Domain and domain-pair ratings dispersion statistics for the Imhonet dataset

| source | target | total KL-divergence | mean user KL-divergence | median user KL-divergence | variance user KL-divergence | source Kurtosis rating | target Kurtosis rating | source skewness rating | target skewness rating |
|---|---|---|---|---|---|---|---|---|---|
| book | game | 6.7790 | 0.0708 | 0.0308 | 0.0139 | 4.8828 | 3.7643 | -1.2867 | -1.1056 |
| book | movie | 0.0300 | 19.0198 | 17.3032 | 155.6620 | 4.2610 | 3.3225 | -1.1816 | -0.8741 |
| book | perfume | 6.3087 | 0.1230 | 0.0664 | 0.0567 | 5.1163 | 3.0342 | -1.3706 | -0.8117 |
| game | book | 1.9261 | 0.0166 | 0.0064 | 0.0014 | 3.7643 | 4.8828 | -1.1056 | -1.2867 |
| game | movie | 1.8925 | 0.0650 | 0.0200 | 0.0150 | 3.6669 | 3.5379 | -1.0931 | -0.8978 |
| game | perfume | 0.0315 | 0.0620 | 0.0143 | 0.0429 | 3.8489 | 3.2307 | -1.0844 | -0.9086 |
| movie | book | 0.0295 | 17.8306 | 10.0413 | 212.1315 | 3.3225 | 4.2610 | -0.8741 | -1.1816 |
| movie | game | 7.9774 | 0.0655 | 0.0299 | 0.0113 | 3.5379 | 3.6669 | -0.8978 | -1.0931 |
| movie | perfume | 7.7443 | 0.1199 | 0.0655 | 0.0501 | 3.5520 | 3.0251 | -0.9039 | -0.8125 |
| perfume | book | 1.5688 | 0.0265 | 0.0106 | 0.0035 | 3.0342 | 5.1163 | -0.8117 | -1.3706 |
| perfume | game | 0.0276 | 0.0476 | 0.0113 | 0.0188 | 3.2307 | 3.8489 | -0.9086 | -1.0844 |
| perfume | movie | 1.5792 | 0.1053 | 0.0349 | 0.0369 | 3.0251 | 3.5520 | -0.8125 | -0.9039 |

Table 51: Domain-pair CCA statistics for the Imhonet dataset

| source | target | CCA $\geq 0.80$ | CCA $\geq 0.90$ | CCA $\geq 0.95$ | average correlation | first component correlation | first 5 components correlation | # components | # significant correlations |
|---|---|---|---|---|---|---|---|---|---|
| book | game | 1 | 0.2 | 0 | 0.8738 | 0.9362 | 0.8738 | 5 | 0 |
| book | movie | 0.8 | 0.2 | 0 | 0.8547 | 0.9354 | 0.8547 | 5 | 0 |
| book | perfume | 1 | 1 | 0.4 | 0.9454 | 0.9820 | 0.9454 | 5 | 0 |
| game | book | 0.8 | 0.2 | 0 | 0.8639 | 0.9244 | 0.8639 | 5 | 0 |
| game | movie | 0.6 | 0.2 | 0 | 0.8308 | 0.9132 | 0.8308 | 5 | 0 |
| game | perfume | 1 | 1 | 0.4 | 0.9421 | 0.9649 | 0.9421 | 5 | 0 |
| movie | book | 0.8 | 0.2 | 0 | 0.8564 | 0.9386 | 0.8564 | 5 | 0 |
| movie | game | 0.6 | 0.2 | 0 | 0.8354 | 0.9111 | 0.8354 | 5 | 0 |
| movie | perfume | 1 | 0.8 | 0.4 | 0.9362 | 0.9852 | 0.9362 | 5 | 0 |
| perfume | book | 1 | 1 | 0.6 | 0.9472 | 0.9764 | 0.9472 | 5 | 0 |
| perfume | game | 1 | 1 | 0.4 | 0.9459 | 0.9660 | 0.9459 | 5 | 0 |
| perfume | movie | 1 | 0.8 | 0.4 | 0.9350 | 0.9842 | 0.9350 | 5 | 0 |

## C.2 ERROR OF ALGORITHMS ON DOMAIN PAIRS IN IMHONET DATASET

Table 52: RMSE and MAE for domain-pairs in the Imhonet dataset

| source | target | CCA RMSE | CD-SVD RMSE | SD-SVD RMSE | CCA MAE | CD-SVD MAE | SD-SVD MAE |
|---|---|---|---|---|---|---|---|
| book | game | 0.2662 | 0.4262 | 0.3336 | 0.2033 | 0.3276 | 0.2464 |
| book | movie | 0.2319 | 0.3245 | 0.3280 | 0.1778 | 0.2514 | 0.2530 |
| book | perfume | 0.2336 | 0.4861 | 0.4137 | 0.1875 | 0.3885 | 0.3240 |
| game | book | 0.2101 | 0.4118 | 0.3717 | 0.1513 | 0.3147 | 0.2808 |
| game | movie | 0.2204 | 0.4295 | 0.4508 | 0.1688 | 0.3419 | 0.3588 |
| game | perfume | 0.2314 | 0.3477 | 0.3489 | 0.1863 | 0.2698 | 0.2710 |
| movie | book | 0.2220 | 0.3027 | 0.3017 | 0.1649 | 0.2262 | 0.2226 |
| movie | game | 0.2300 | 0.4388 | 0.3545 | 0.1756 | 0.3333 | 0.2656 |
| movie | perfume | 0.2369 | 0.4536 | 0.3667 | 0.1891 | 0.3612 | 0.2859 |
| perfume | book | 0.2032 | 0.4116 | 0.3726 | 0.1492 | 0.3100 | 0.2753 |
| perfume | game | 0.2118 | 0.3082 | 0.3070 | 0.1654 | 0.2328 | 0.2296 |
| perfume | movie | 0.2131 | 0.4349 | 0.4188 | 0.1641 | 0.3457 | 0.3324 |

## C.3 COLD-START ANALYSIS OF DOMAIN PAIRS FOR IMHONET DATASET

### C.3.1 MAEs for Target User Profiles

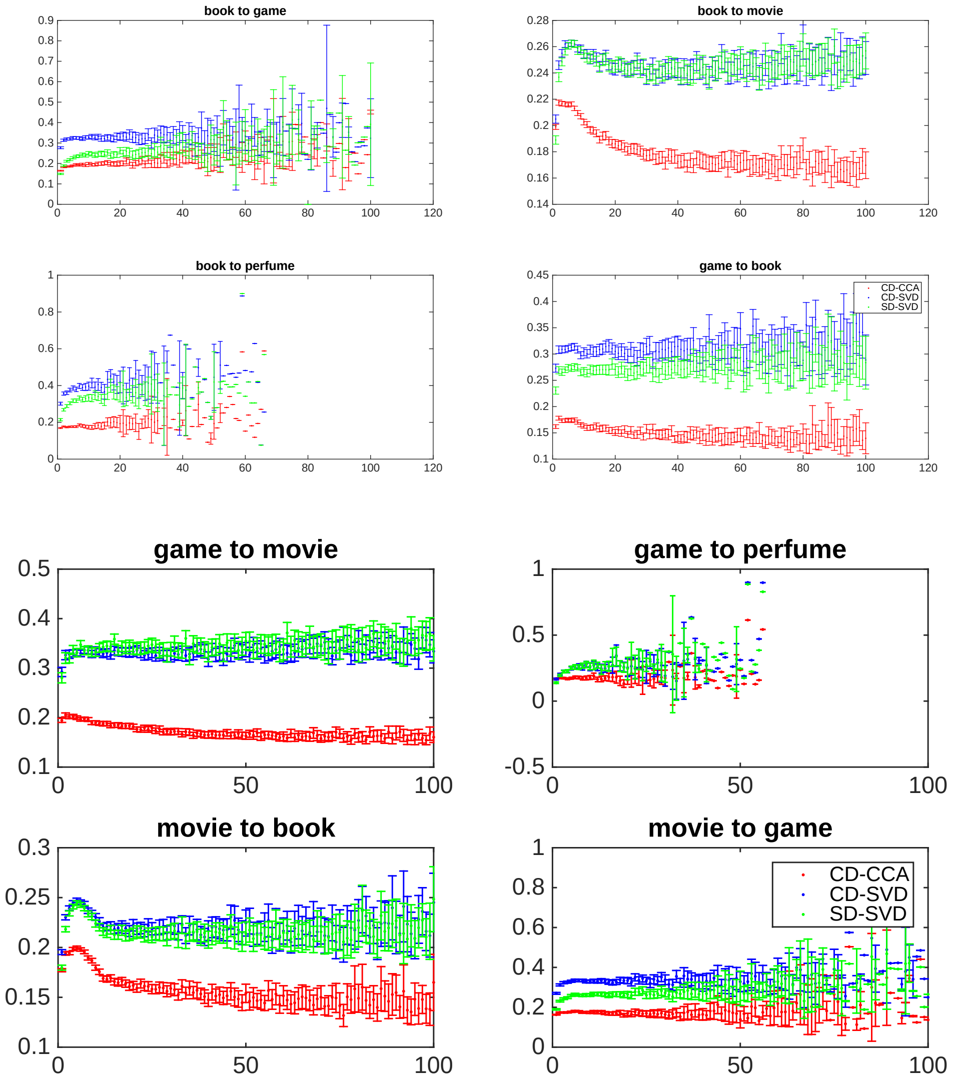


Figure 97: User-based MAE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' target domain profile size

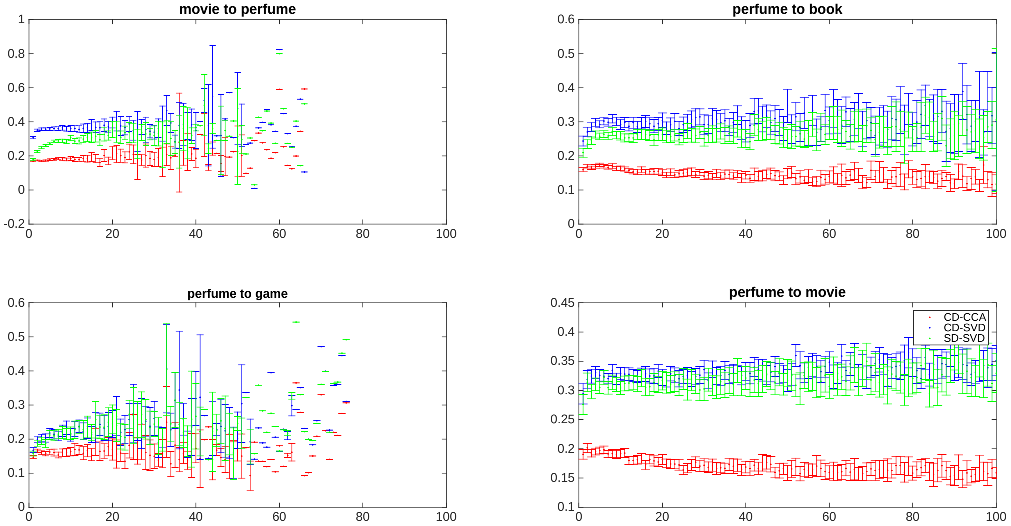


Figure 97: User-based MAE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 9 to 12

### C.3.2 RMSEs for Target User Profiles

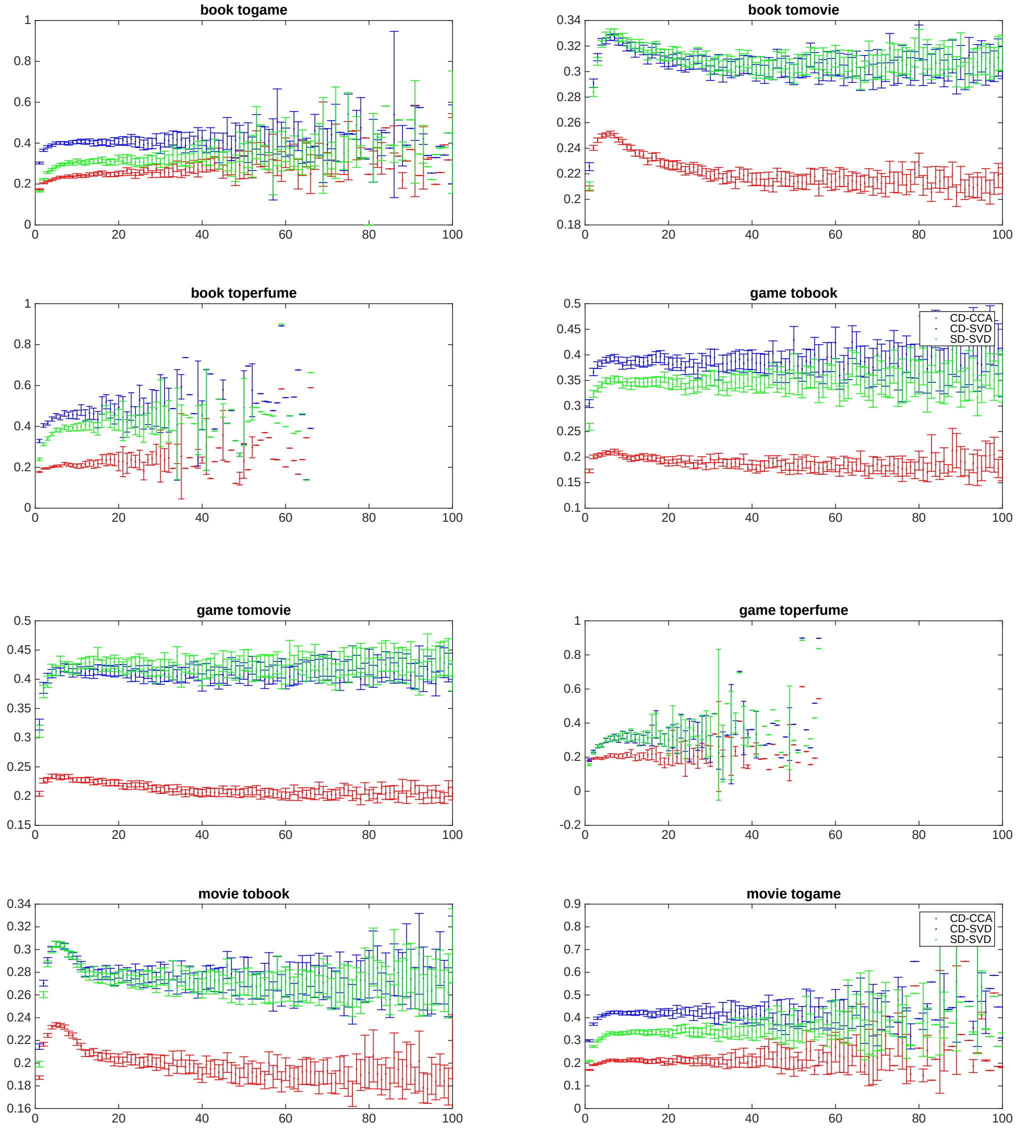


Figure 98: User-based RMSE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' target domain profile size

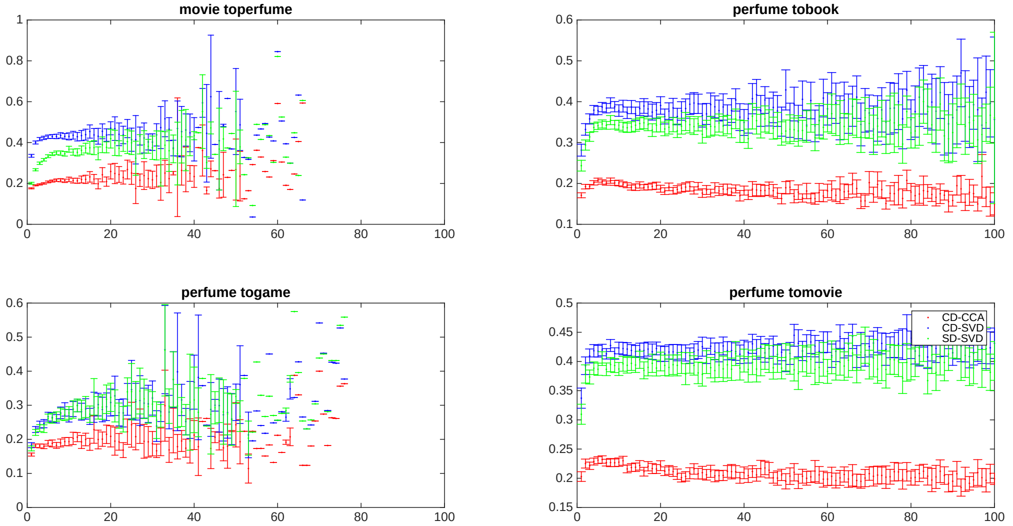


Figure 98: User-based RMSE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' target domain profile size for domain pairs 9 to 12

### C.3.3 MAEs for Source User Profiles

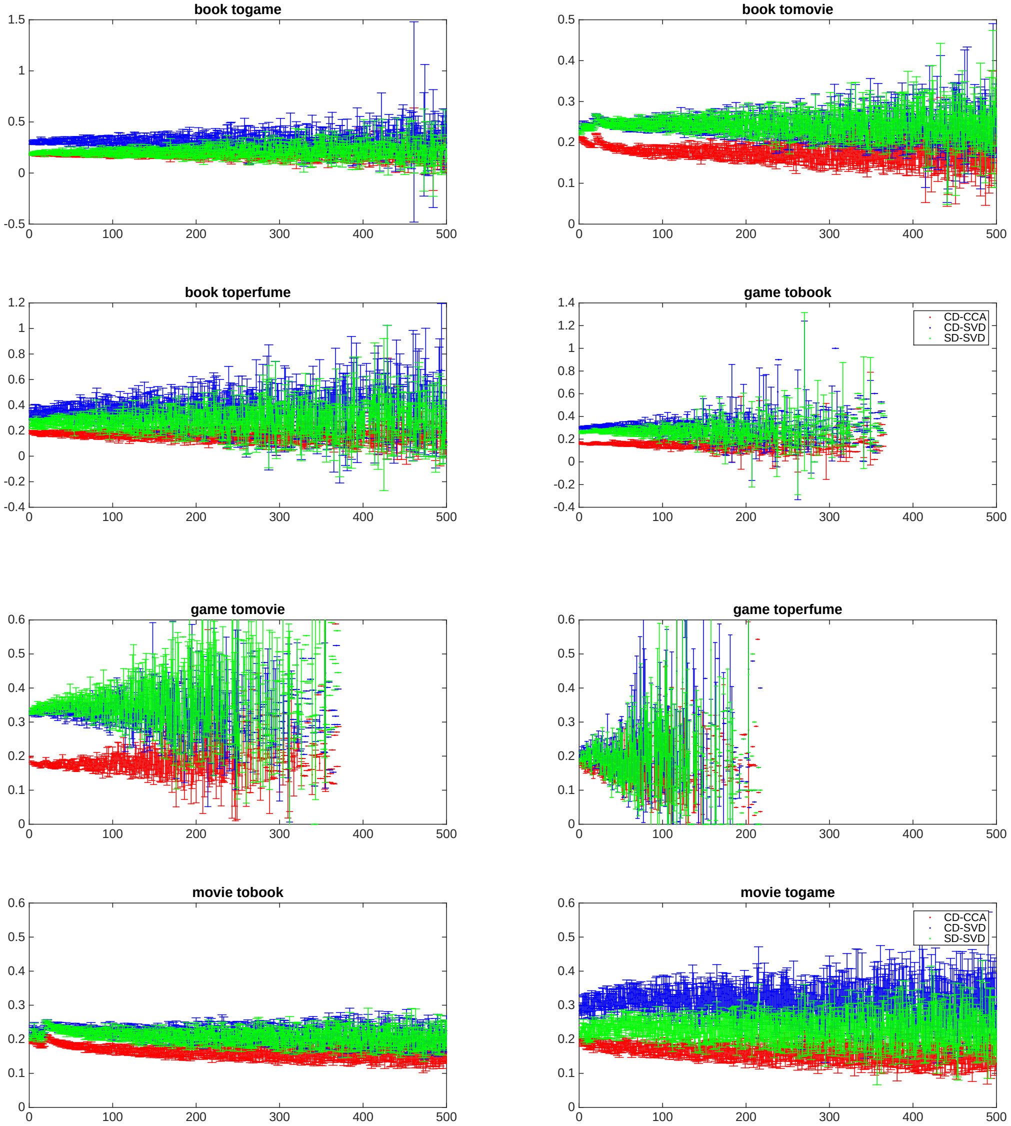


Figure 99: User-based MAE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size

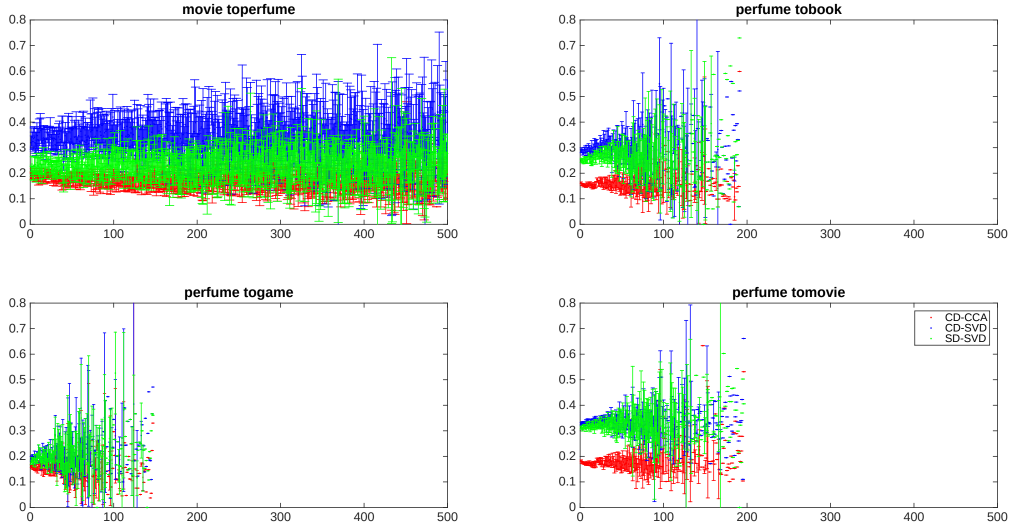


Figure 99: User-based MAE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size for domain pairs 9 to 12

### C.3.4 RMSEs for Source User Profiles

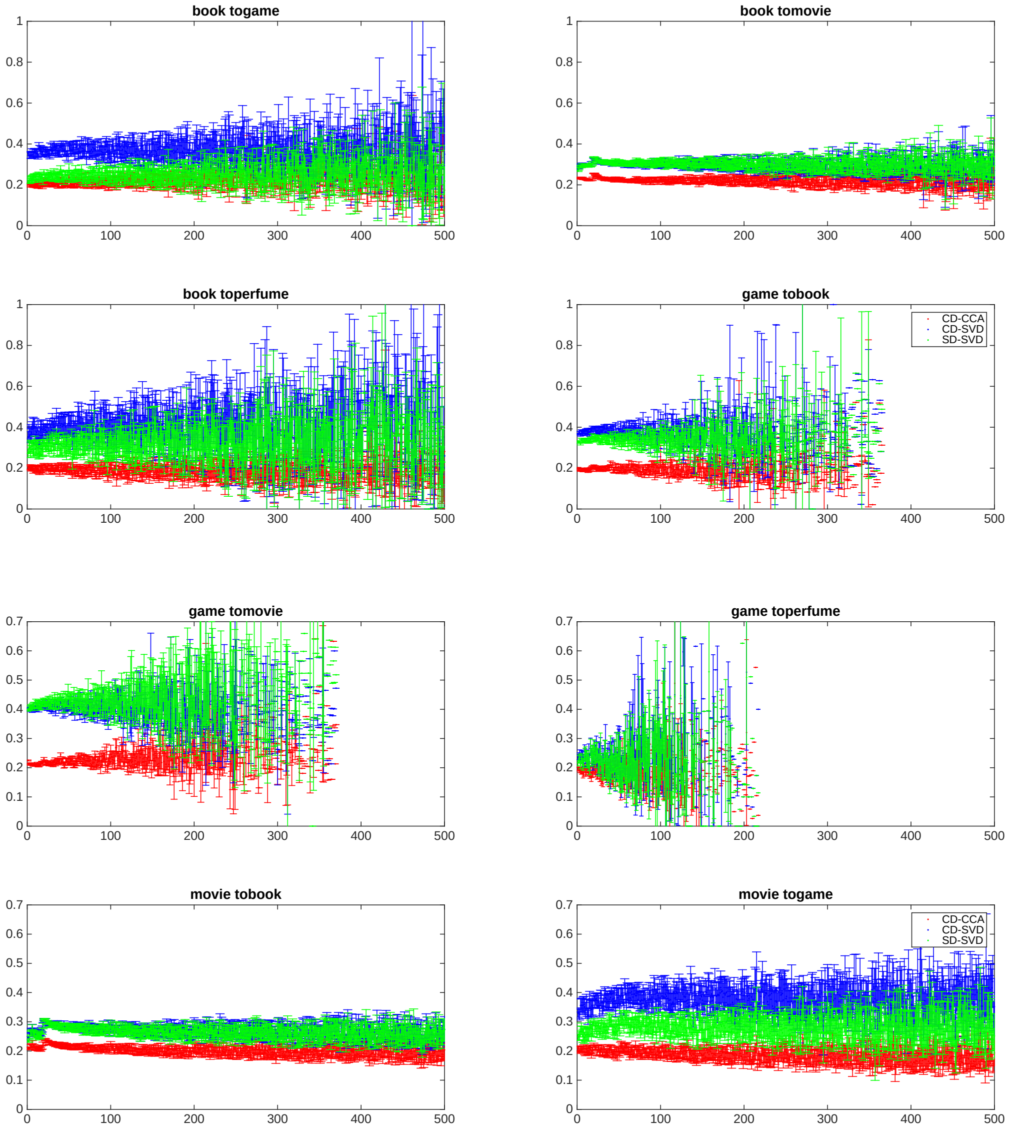


Figure 100: User-based RMSE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size

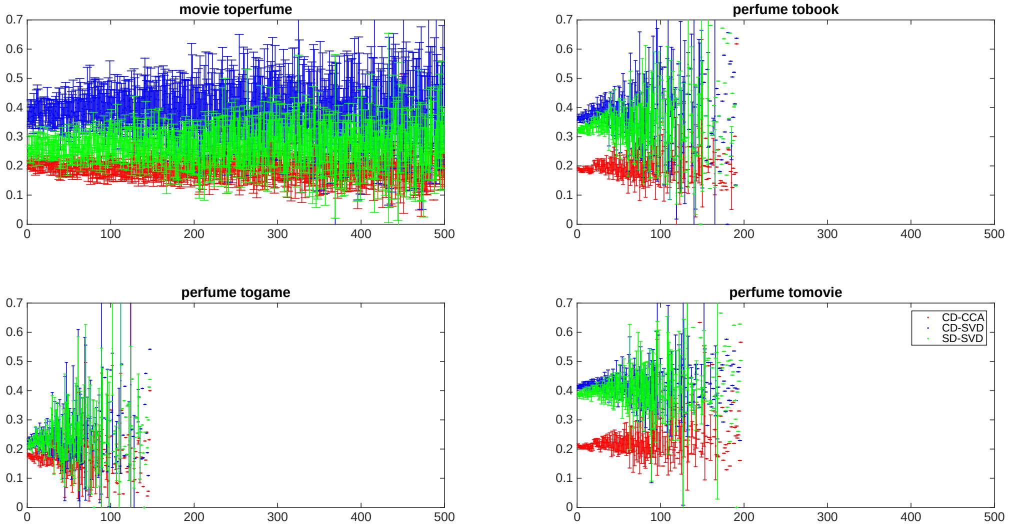


Figure 100: User-based RMSE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' Source domain profile size for domain pairs 9 to 12

## C.4 CORRELATION ANALYSIS FOR IMHONET DATASET

### C.4.1 Correlation Analysis Plots for the RMSE of Algorithms in the Imhonet Dataset

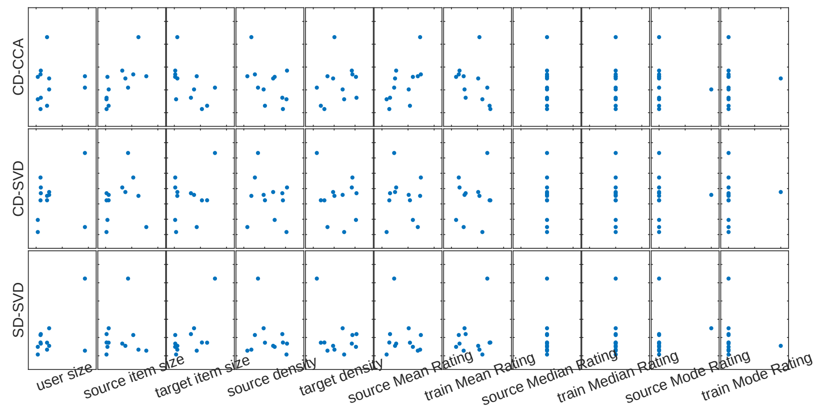


Figure 101: Scatter plot of general and central tendency statistics with the RMSE of algorithms in the Imhonet dataset

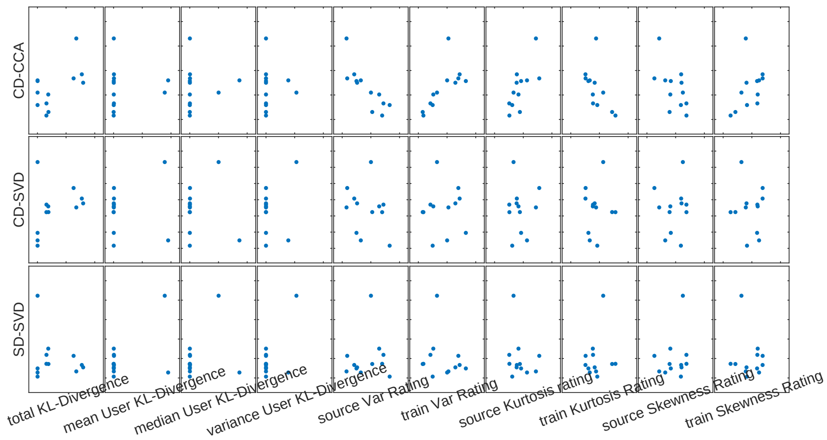


Figure 102: Scatter plot of dispersion statistics with the RMSE of algorithms in the Imhonet dataset

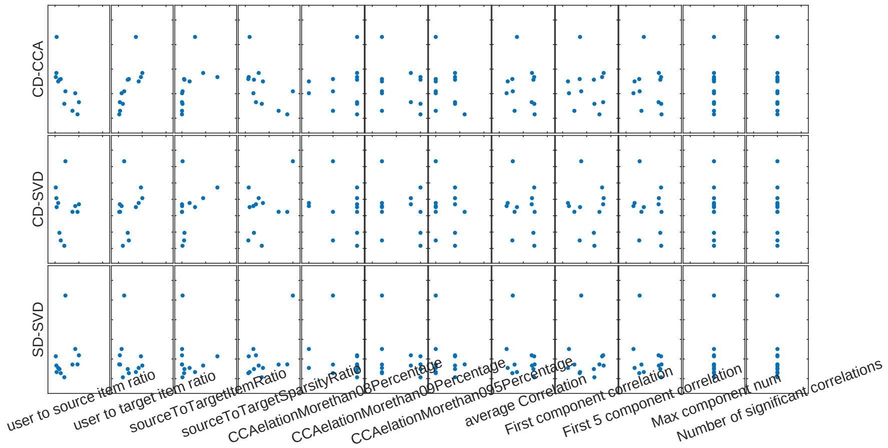


Figure 103: Scatter plot of CCA-related statistics with the RMSE of algorithms in the Imhonet dataset

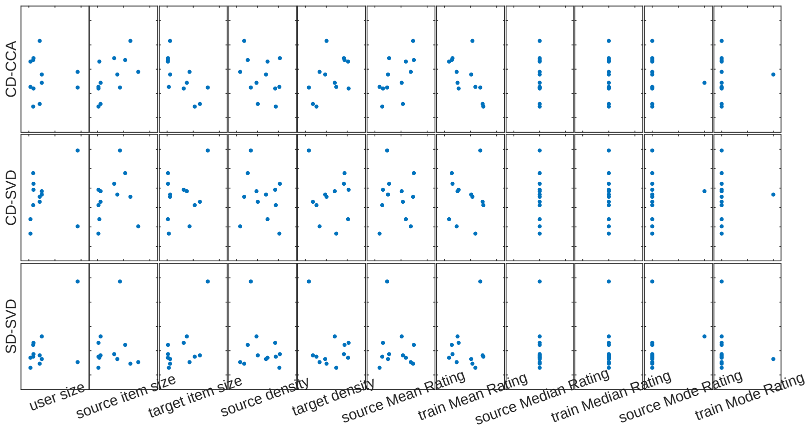


Figure 104: Scatter plot of general and central tendency statistics with the MAE of algorithms in the Imhonet dataset

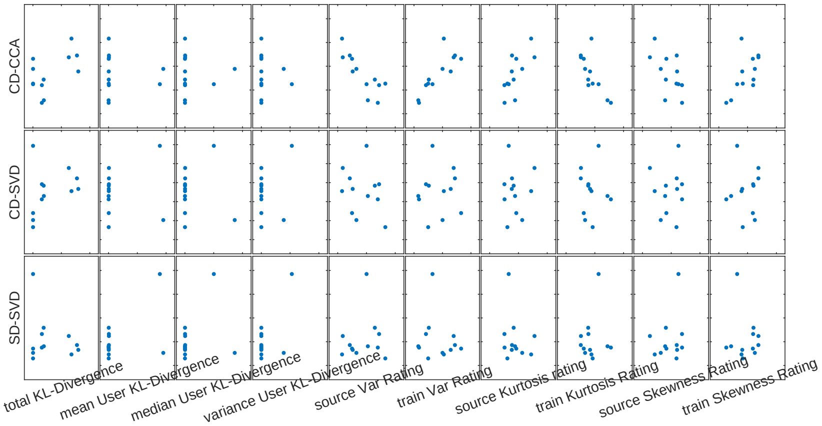


Figure 105: Scatter plot of dispersion statistics with the MAE of algorithms in the Imhonet dataset

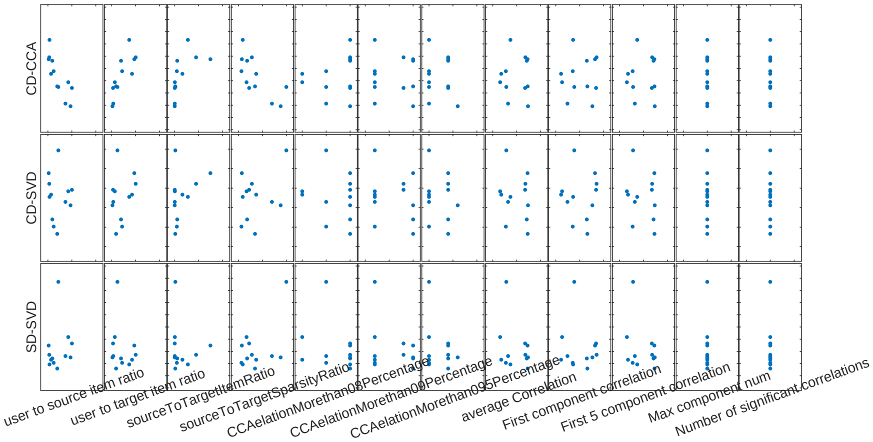


Figure 106: Scatter plot of CCA-related statistics with the MAE of algorithms in the Imhonet dataset

### C.4.2 Correlation Analysis Plots for the Improvement Ratio of Algorithms in the Imhonet Dataset

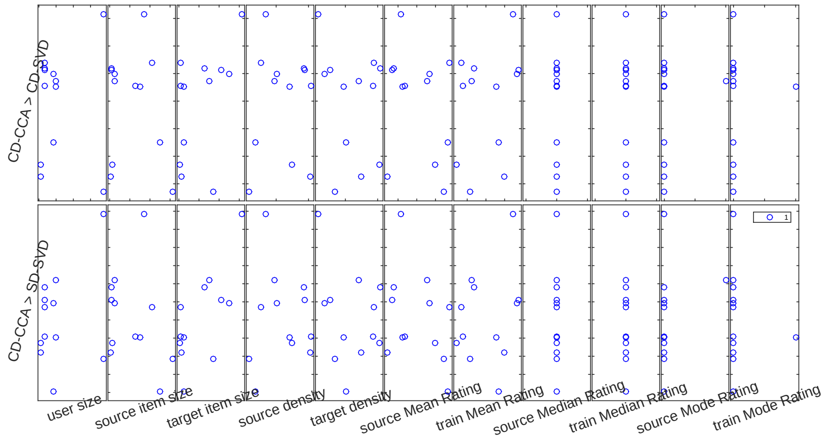


Figure 107: Scatter plot of general and central tendency statistics with the improvement ratio of CD-CCA over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms

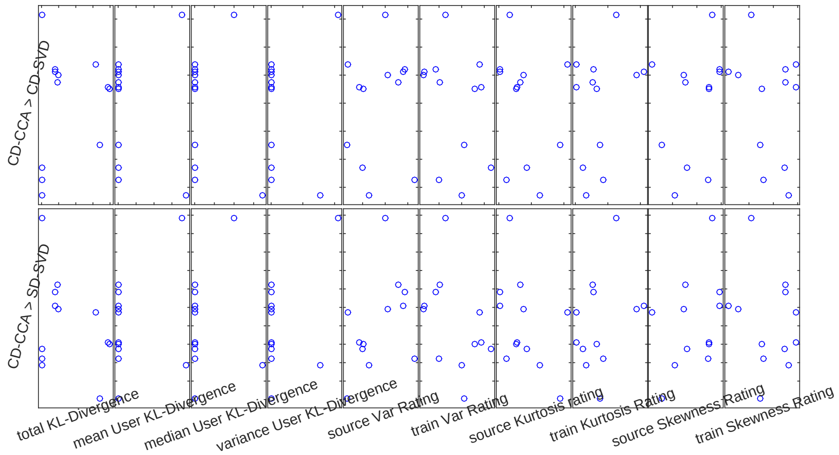


Figure 108: Scatter plot of dispersion statistics with the improvement ratio of CD-CCA over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms

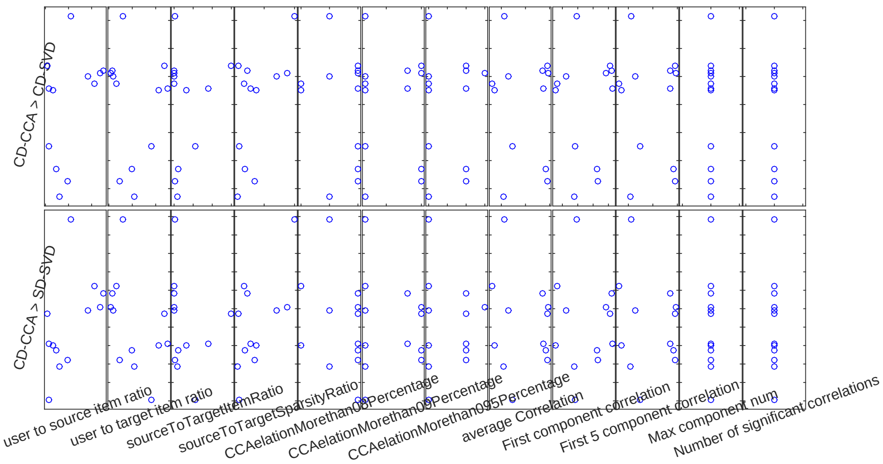


Figure 109: Scatter plot of CCA-related statistics with the improvement ratio of CD-CCA over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-CCA is significantly better than other algorithms

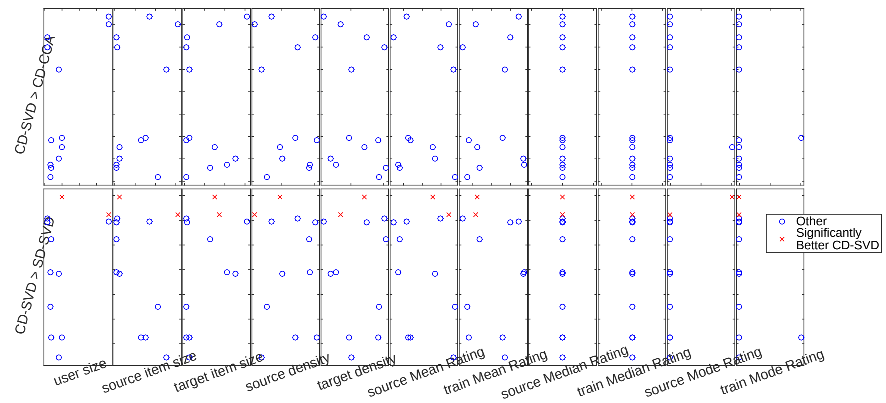


Figure 110: Scatter plot of general and central tendency statistics with the improvement ratio of CD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms

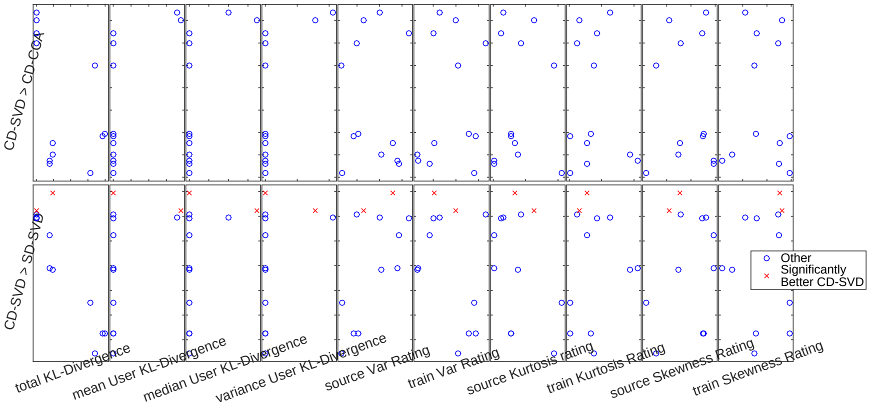


Figure 111: Scatter plot of dispersion statistics with the improvement ratio of CD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms

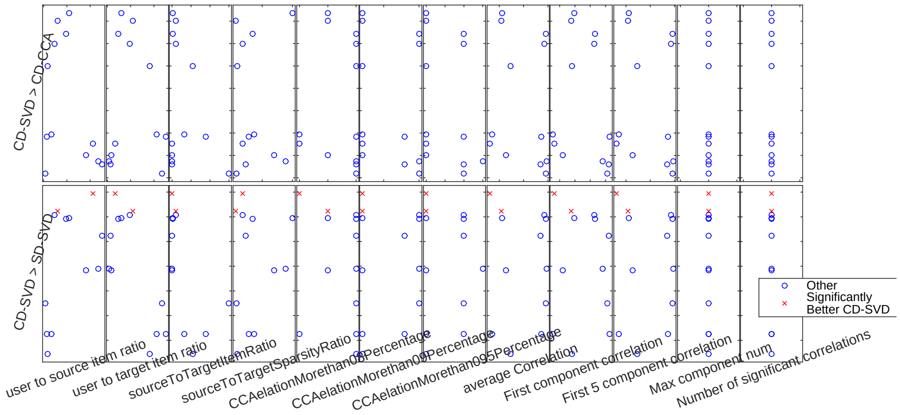


Figure 112: Scatter plot of CCA-related statistics with the improvement ratio of CD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which CD-SVD is significantly better than other algorithms

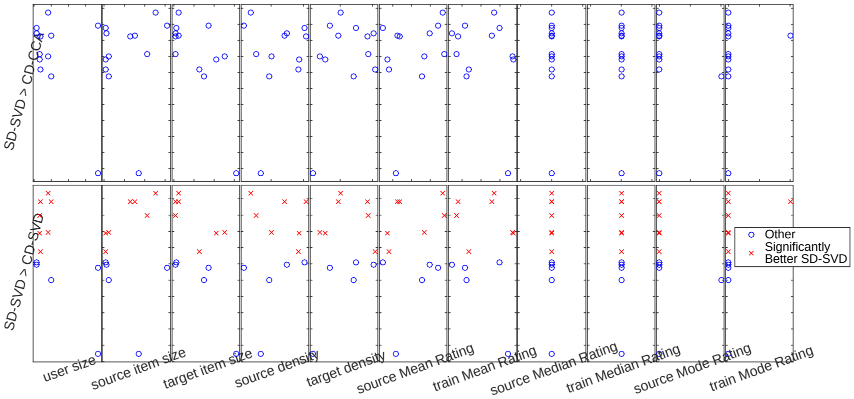


Figure 113: Scatter plot of general and central tendency statistics with the improvement ratio of SD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms

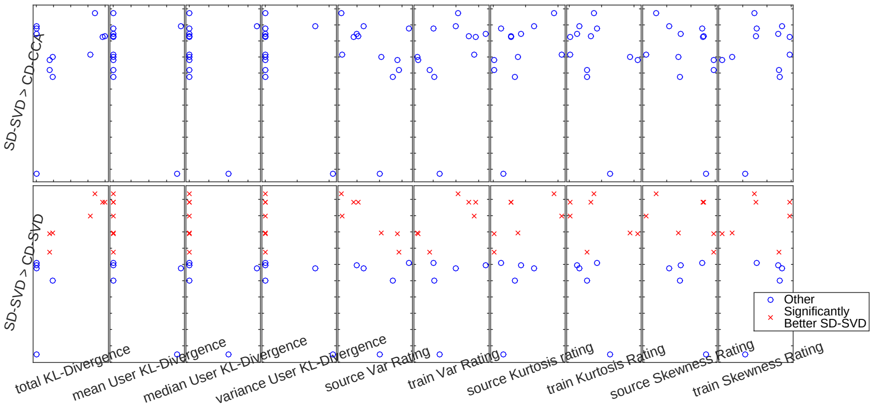


Figure 114: Scatter plot of dispersion statistics with the improvement ratio of SD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms

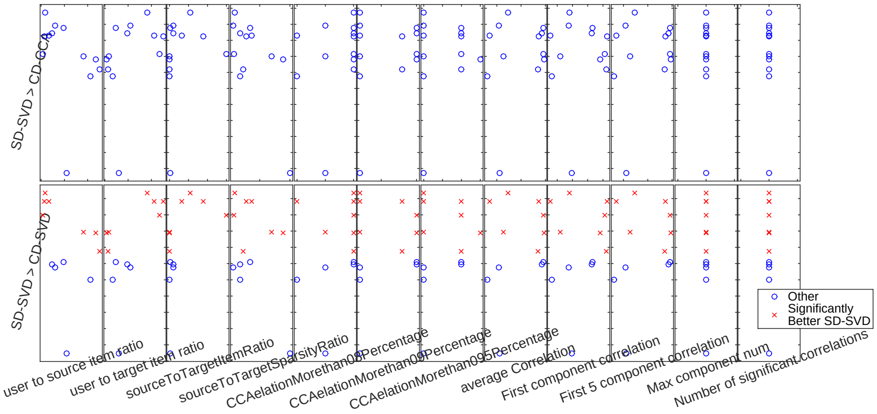


Figure 115: Scatter plot of CCA-related statistics with the improvement ratio of SD-SVD over other algorithms in the Imhonet dataset; red cross shows the cases in which SD-SVD is significantly better than other algorithms

# BIBLIOGRAPHY

[1] E. Acar, T. G. Kolda, and D. M. Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. *arXiv preprint arXiv:1105.3422*, 2011.

[2] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1):103–145, 2005.

[3] G. Adomavicius and A. Tuzhilin. Multidimensional recommender systems: a data warehousing approach. In *Electronic commerce*, pages 180–192. Springer Berlin Heidelberg, 2001.

[4] S. Akaho. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.

[5] D. A. Belsley, E. Kuh, and R. E. Welsch. *Regression diagnostics: Identifying influential data and sources of collinearity*, volume 571. John Wiley & Sons, 2005.

[6] S. Berkovsky, T. Kuflik, and F. Ricci. Cross-domain mediation in collaborative filtering. In *Proceedings of the 11th international conference on User Modeling*, UM '07, Berlin, Heidelberg, 2007. Springer-Verlag.

[7] S. Berkovsky, T. Kuflik, and F. Ricci. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3), Aug. 2008.

[8] R. Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[9] W. Chen, W. Hsu, and M. L. Lee. Making recommendations from multiple domains. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 892–900. ACM, 2013.

[10] P. Cremonesi and M. Quadrana. Cross-domain recommendations without overlapping data: Myth or reality? In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 297–300, New York, NY, USA, 2014. ACM.

[11] P. Cremonesi, A. Tripodi, and R. Turrin. Cross-domain recommender systems. In *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE. Discussed many approaches, but evaluated using articicially separated netflix dataset. Mostly alg comparizon.

[12] Z. Dong and Q. Zhao. Experimental analysis on cross domain preferences association and rating prediction. In *Proceedings of the 1st International Workshop on Cross Domain Knowledge Discovery in Web and Social Network Mining*, pages 26–31. ACM, 2012.

[13] S. Dooms, T. De Pessemier, and L. Martens. Mining cross-domain rating datasets from structured data on twitter. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 621–624. International World Wide Web Conferences Steering Committee, 2014.

[14] M. Eirinaki, C. Lampos, S. Paulakis, and M. Vazirgiannis. Web personalization integrating content semantics and navigational patterns. In *Proceedings of the 6th annual ACM international workshop on Web information and data management*, pages 72–79. ACM, 2004.

[15] A. Elkahky, Y. Song, and X. He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *International World Wide Web Conference Committee (IW3C2)*, 2015.

[16] M. Enrich, M. Braunhofer, and F. Ricci. Cold-start management with cross-domain collaborative filtering and tags. In *E-Commerce and Web Technologies*, pages 101–112. Springer, 2013.

[17] S. Faridani. Using canonical correlation analysis for generalized sentiment analysis, product recommendation and search. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, New York, NY, USA, 2011. ACM.

[18] I. Fernández-Tobías and I. Cantador. Exploiting social tags in matrix factorization models for cross-domain collaborative filtering. In *Proceedings of the 1st Workshop on New Trends in Content-based Recommender Systems, Foster City, California, USA*, pages 34–41, 2014.

[19] I. Fernández-Tobías, I. Cantador, M. Kaminskas, and F. Ricci. A generic semantic-based framework for cross-domain recommendation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, New York, NY, USA, 2011. ACM.

[20] I. Fernández-Tobías, I. Cantador, M. Kaminskas, and F. Ricci. Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*, 2012.

[21] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, and J. Guo. Cross-domain recommendation via cluster-level latent factor model. In *Machine Learning and Knowledge Discovery in Databases*, pages 161–176. Springer, 2013.

[22] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[23] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.

[24] H. Hotelling. Relations Between Two Sets of Variates. *Biometrika*, 28(3/4), 1936.

[25] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 595–606. International World Wide Web Conferences Steering Committee, 2013.

[26] Y.-J. Huang, E. W. Xiang, and R. Pan. Constrained collective matrix factorization. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 237–240. ACM, 2012.

[27] T. Iwata and K. Takeuchi. Cross-domain recommendation without shared users or items by sharing latent vector distributions. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 379–387, 2015.

[28] M. Joshi, M. Dredze, W. W. Cohen, and C. P. Rosé. What's in a domain? multi-domain learning for multi-attribute data. In *Proceedings of NAACL-HLT*, pages 685–690, 2013.

[29] M. Kaminskas and F. Ricci. Location-adapted music recommendation using tags. In *User Modeling, Adaption and Personalization*, pages 183–194. Springer, 2011.

[30] A. Klami, G. Bouchard, and A. Tripathi. Group-sparse embeddings in collective matrix factorization. *arXiv preprint arXiv:1312.5921*, 2013.

[31] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[32] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 173–182. ACM, 2012.

[33] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*, pages 331–339, 1995.

[34] B. Li. Cross-domain collaborative filtering: A brief survey. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pages 1085–1086. IEEE, 2011.

[35] B. Li, Q. Yang, and X. Xue. Can movies and books collaborate?: cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[36] B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 617–624. ACM, 2009.

[37] B. Li, X. Zhu, R. Li, and C. Zhang. Rating knowledge sharing in cross-domain collaborative filtering. *Cybernetics, IEEE Transactions on*, 45(5):1054–1068, 2015.

[38] J. Li and O. R. Zaïane. Combining usage, content, and structure data to improve web site recommendation. In *E-Commerce and Web Technologies*, pages 305–315. Springer, 2004.

[39] H. Lieberman et al. Letizia: An agent that assists web browsing. *IJCAI (1)*, 1995:924–929, 1995.

[40] Q. Liu, S. Wu, and L. Wang. Cot: Contextual operating tensor for context-aware recommender systems. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[41] Y.-F. Liu, C.-Y. Hsu, and S.-H. Wu. Non-linear cross-domain collaborative filtering via hyper-structure transfer. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1190–1198, 2015.

[42] B. Loni, Y. Shi, M. Larson, and A. Hanjalic. Cross-domain collaborative filtering with factorization machines. In *Advances in Information Retrieval*, pages 656–661. Springer, 2014.

[43] Y. Low, D. Agarwal, and A. J. Smola. Multiple domain user personalization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 123–131. ACM, 2011.

[44] Y. Lu and D. P. Foster. large scale canonical correlation analysis with iterative least squares. In *Advances in Neural Information Processing Systems*, pages 91–99, 2014.

[45] Z. Lu, W. Pan, E. W. Xiang, Q. Yang, L. Zhao, and E. Zhong. Selective transfer learning for cross domain recommendation. In *SDM*, pages 641–649. SIAM, 2013.

[46] N. Mirbakhsh and C. X. Ling. Improving top-n recommendation for cold-start users via cross-domain information. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(4):33, 2015.

[47] B. Mobasher. Data mining for web personalization. In *The adaptive web*, pages 90–135. Springer, 2007.

[48] B. Mobasher, H. Dai, T. Luo, Y. Sun, and J. Zhu. Integrating web usage and content mining for more effective personalization. In *Electronic commerce and web technologies*, pages 165–176. Springer, 2000.

[49] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204. ACM, 2000.

[50] O. Moreno, B. Shapira, L. Rokach, and G. Shani. Talmud: transfer learning for multiple domains. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 425–434. ACM, 2012.

[51] H. Ohkushi, T. Ogawa, and M. Haseyama. Kernel cca-based music recommendation according to human motion robust to temporal expansion. In *Communications and Information Technologies (ISCIT), 2010 International Symposium on*, oct. 2010.

[52] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*, 2010.

[53] W. Pan, E. W. Xiang, and Q. Yang. Transfer learning in collaborative filtering with uncertain ratings. In *AAAI*, 2012.

[54] W. Pan and Q. Yang. Transfer learning in heterogeneous collaborative filtering domains. *Artificial intelligence*, 197:39–55, 2013.

[55] R. Parimi and D. Caragea. Cross-domain matrix factorization for multiple implicitfeedback domains. In *International Workshop on Machine learning, Optimization and big Data (MOD)(Accepted)*, 2015.

[56] D. Parra and S. Sahebi. Recommender systems: Sources of knowledge and evaluation metrics. In *Advanced Techniques in Web Intelligence-2*, pages 149–175. Springer, 2013.

[57] M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.

[58] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[59] S. D. Roy, T. Mei, W. Zeng, and S. Li. Empowering cross-domain internet media with real-time topic learning from social streams. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 49–54. IEEE, 2012.

[60] S. Sahebi and P. Brusilovsky. Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In *User Modeling, Adaptation, and Personalization*, pages 289–295. Springer, 2013.

[61] S. Sahebi and W. Cohen. Community-based recommendations: a solution to the cold start problem. In *Workshop on Recommender Systems and the Social Web, in conjunction with ACM RecSys'11*. ACM, 2011.

[62] S. Sahebi and T. Walker. Content-based cross-domain recommendations using segmented models. *CBRecSys 2014*, page 57, 2014.

[63] J. B. Schafer, J. Konstan, and J. Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166. ACM, 1999.

[64] B. Shapira, L. Rokach, and S. Freilikhman. Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, 23(2-3):211–247, 2013.

[65] J. Shi, M. Long, Q. Liu, G. Ding, and J. Wang. Twin bridge transfer learning for sparse collaborative filtering. In *Advances in Knowledge Discovery and Data Mining*, pages 496–507. Springer, 2013.

[66] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658. ACM, 2008.

[67] L. Sun, S. Ji, and J. Ye. A least squares formulation for canonical correlation analysis. In *Proceedings of the 25th international conference on Machine learning*, pages 1024–1031. ACM, 2008.

[68] S. Tan, J. Bu, X. Qin, C. Chen, and D. Cai. Cross domain recommendation based on multi-type media fusion. *Neurocomputing*, 127:124–134, 2014.

[69] S. Tantanasiriwong and C. Haruechaiyasak. Cross-domain citation recommendation based on co-citation selection. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014 11th International Conference on*, pages 1–4. IEEE, 2014.

[70] A. Tiroshi and T. Kuflik. Domain ranking for cross domain collaborative filtering. In *User Modeling, Adaptation, and Personalization*, pages 328–333. Springer, 2012.

[71] A. Umyarov and A. Tuzhilin. Leveraging aggregate ratings for better recommendations. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 161–164, New York, NY, USA, 2007. ACM.

[72] A. Umyarov and A. Tuzhilin. Improving rating estimation in recommender systems using aggregation- and variance-based hierarchical models. In *Proceedings of the Third*

*ACM Conference on Recommender Systems*, RecSys '09, pages 37–44, New York, NY, USA, 2009. ACM.

[73] A. Umyarov and A. Tuzhilin. Using external aggregate ratings for improving individual recommendations. *ACM Transactions on the Web (TWEB)*, 5(1):3, 2011.

[74] W. Wang, Z. Chen, J. Liu, Q. Qi, and Z. Zhao. User-based collaborative filtering on cross domain by tag transfer learning. In *Proceedings of the 1st International Workshop on Cross Domain Knowledge Discovery in Web and Social Network Mining*, pages 10–17. ACM, 2012.

[75] D. Weenink. Canonical correlation analysis. In *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*, volume 25, pages 81–99, 2003.

[76] P. Winoto and T. Tang. If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations. *New Generation Computing*, 26, 2008. 10.1007/s00354-008-0041-0.

[77] L. Wu, W. Zhang, and J. Wang. Fusion hidden markov model with latent dirichlet allocation model in heterogeneous domains. In *Proceedings of International Conference on Internet Multimedia Computing and Service*, page 261. ACM, 2014.

[78] X. Xin, Z. Liu, and H. Huang. A nonlinear cross-site transfer learning approach for recommender systems. In *Neural Information Processing*, pages 495–502. Springer, 2014.

[79] X. Yang, T. Zhang, and C. Xu. Cross-domain feature learning in multimedia. *IEEE Transactions on Multimedia*, 17:64 – 78, 2015.

[80] C. Yi, M.-S. Shang, and Q.-M. Zhang. Auxiliary domain selection in cross-domain collaborative filtering. *Appl. Math*, 9(3):1375–1381, 2015.

[81] Y. Zhang. Browser-oriented universal cross-site recommendation and explanation based on user browsing logs. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 433–436. ACM, 2014.

[82] Y. Zhang, B. Cao, and D.-Y. Yeung. Multi-domain collaborative filtering. *arXiv preprint arXiv:1203.3535*, 2012.

[83] L. Zhao, S. J. Pan, E. W. Xiang, E. Zhong, Z. Lu, and Q. Yang. Active transfer learning for cross-system recommendation. In *AAAI*. Citeseer, 2013.