Active Learning of Classification Models from Enriched Label-related Feedback

by

Yanbing Xue

Bachelor of Engineering, Shandong University, 2013

Master of Science, University of Pittsburgh, 2015

Submitted to the Graduate Faculty of

the Dietrich School of Art and Science in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2020

Active Learning of Classification Models from Enriched Label-related Feedback

Yanbing Xue, PhD

University of Pittsburgh, 2020

Our ability to learn accurate classification models from data is often limited by the number of available labeled data instances. This limitation is of particular concern when data instances need to be manually labeled by human annotators and when the labeling process carries a significant cost. Recent years witnessed increased research interest in developing methods in different directions capable of learning models from a smaller number of examples. One such direction is active learning, which finds the most informative unlabeled instances to be labeled next. Another, more recent direction showing a great promise utilizes enriched label-related feedback. In this case, such feedback from the human annotator provides additional information reflecting the relations among possible labels. The cost of such feedback is often negligible compared with the cost of instance review. The enriched label-related feedback may come in different forms. In this work, we propose, develop and study classification models for binary, multi-class and multi-label classification problems that utilize the different forms of enriched label-related feedback. We show that this new feedback can help us improve the quality of classification models compared with the standard class-label feedback. For each of the studied feedback forms, we also develop new active learning strategies for selecting the most informative unlabeled instances that are compatible with the respective feedback form, effectively combining two approaches for reducing the number of required labeled instances. We demonstrate the effectiveness of our new framework on both simulated and real-world datasets.

Keywords: active learning, classification, multi-class, multi-label, enriched label-related feedback, probabilistic score, Likert-scale feedback, ordered class set.

Table of Contents

List of Tables

List of Figures

Preface

I would like to thank everybody who made my journey to Ph.D. degree possible and joyful.

First of all, I want to express my deepest gratitude to my advisor, Prof. Milos Hauskrecht, who has always supported me in both professional career and personal life. Milos taught me machine learning and data mining, and how to conduct high quality research in these fields. He also encouraged and guided me through difficult moments during my graduate school years in +Pittsburgh.

I would like to thank my thesis committee, Prof. Diane Litman, Prof. Adriana Kovashka, and Prof. Shyam Visweswaran, for their valuable feedback and discussions on my thesis research. Parts of this thesis are the results of collaboration with colleagues from University of Pittsburgh. In particular, I want to thank our fellow doctorate student, Zhipeng Luo (currently at Northwest Univ.), who helped me on several papers and gave me insights in some machine learning and optimization techniques. I also would like to thank all the other former and current members of Milos' lab: Siqi Liu, Jeongmin Lee, Matt Barren, Dr. Charmgil Hong (currently at Handong Univ.), Dr. Zitao Liu (currently at TAL education group), Dr. Mahdi Pakdaman (currently at Microsoft), Dr. Eric Heim (currently at Carnegie Mellon Univ.), and Dr. Quang Nguyen (currently at Intuit).

I am also grateful to have many good friends in Pittsburgh, who made my life more enjoyable. In particular, I want to mention my best friend Xiaoyu Ge, who was always willing to help and gave me a lot of useful advices.

Finally, I am very grateful to my parents Wei and Shujian, my grandparents, and my cousins for their unlimited love, encouragement and support.

Thank you all!

1.0 Introduction

In recent years, the world has witnessed a remarkable increase in the number and quality of classification models built from data. One important factor contributing to this improvement is the number of labeled data instances available to train these models. However, this improvement may not be possible when the original data are unlabeled or when the labels are obtained through additional human annotation effort. Take for example a patient's health record. While some of the data (e.g., medications given, lab tests) are recorded, diagnoses of some conditions or adverse events that occurred during the hospitalization are not. In order to analyze and predict these conditions, individual patient instances must be first labeled by an expert. However, the process of labeling instances using subjective human assessments faces the following problem:

Collecting labels from human annotators can be extremely time-consuming and costly, especially in the domains where data assessments require a high level of expertise. In disease diagnostics, an experienced physician needs to spend about five minutes (on average) to review and evaluate one patient case [Nguyen et al., 2011a]; or in speech recognition, [Zhu, 2005] reports that a trained linguist may take up to seven hours to annotate one minute of audio record (e.g., 400 times as long). The challenge is to find ways to reduce the number of instances that need to be reviewed and labeled by an expert while improving the quality of the models learned from these instances. In this thesis, we study two complementary approaches for achieving this goal and for reducing the human annotation effort: enriched label-related feedback and active learning.

The text in the remainder of this chapter is organized as follows. We first review the different types of human feedback one can use to build classification models and introduce the enriched label-related feedback studied throughout the thesis. After that we discuss active learning strategies tailored to optimize the benefits of enriched label feedback aimed to further reduce the instance annotation cost. These two methods and their annotation benefits are the centerpieces of the main hypothesis of the thesis and its application to three classification problems: binary, multi-class and multi-label classification. Finally, we give a roadmap to the chapters of the thesis by introducing the key problems they aim to solve.

1.1 Building a classification model from human feedback

Our objective is to build a classification model. Let us assume we do not have labeled data instances to accomplish this task. What is it we can do? It is clear we need some help/feedback from a human to build a model. However, this feedback may come in different forms and with different advantages and shortcomings, say, at different cost for feedback provision. We divide the types of feedback the human experts may provide to build a classification model into three categories: (1) Model-based, (2) Instance-based, (3) Structure-based feedback.

- Model-based feedback relies on a human expert to define and build a complete classification model based on their knowledge. Typically the model is defined in some knowledge-representation language that is easy to use for a human, such as rules [Zhao et al., 2009] or decision trees [Tung, 2009]. The model-based approach expects the expert to select both the input features to use in the model and thresholds on the features. The advantage of model-based methods is their ability to handle high-dimensional data with many irrelevant features, as these are explicitly filtered out by humans. The limitation is that the burden of defining the model is squarely on the shoulders of the expert, the computational methods or tools only try to facilitate the process. Most often the hardest part of the model building process is parameter tuning. In real-world classification tasks, especially tasks with continuously numerical features, where the thresholds are not clear-cut values, the tuning of these by human experts typically requires search and iterative refinement of the model, which may significantly increase the cost.
- Instance-based feedback relies on a human to annotate a collection of prototypes used to describe the dataset. A prototype might be just one data instance, or some hypothetical instance computed from one or more of them (such as the weighted average of a set of instances). A new instance is classified by finding similar prototypes and using their classes in some way to form a prediction. In other words, human experts just need to provide the classes of some prototypes, where each prototype can be generated from one or more data instances, and the machine learning researchers are responsible to infer the implicit generalization of the dataset based on the classes of these prototypes. Enriched label-related feedback, the direction we explore in this thesis, is a subset of instance-based methods. The main advantage of instance-based methods is

ability of parameter tuning. It is easy to find the optimal parameters simply by maximizing the accuracy or other desired evaluation metrics on the prototypes provided by the human experts. The main disadvantage is the amount of prototypes needed for obtaining a classification model, especially for a high-dimensional dataset. Model-based methods may eliminate the irrelevant features simply from the knowledge of the human experts, while instance-based methods may require many prototypes to infer such implicit irrelevancy.

- Structure-based feedback is an eclectic category in between pure model-based and instance-based methods. We use this category to cover human feedback that is not sufficient on its own to build a classification model but can still significantly aid the process. Briefly, like model-based methods, the human experts are asked to provide the information on the classification model. However, only some properties or structures of the model, typically the correlations among relevant features and class labels, or rough values of some parameters are provided based on the knowledge of the human experts. Hence the structure-based feedback needs to be combined with other knowledge or data to complete the model building process. An example of a structure based feedback is feature-based feedback [Druck et al., 2009]. It helps the model building processes by defining and selecting the input features to use in the model. It is one of the steps in the model-based approach, and by itself, it is not sufficient to define the full classification model so it needs to be combined with other approaches and other types of feedback to finalize the model building process. The benefit of the feature feedback is that it helps to narrow down the input features to use in the model, and as a result, it is very useful when data are high-dimensional and when they include many features irrelevant for the classification task. Another example of the structure-based feedback is a methods proposed by [Collins, 2003] for part-of-speech (POS) tagging in natural language processing, where each word in a sentence is assigned a POS tag (class) indicating its grammatical role in this sentence. This method asked the human experts to provide a collection of grammar trees representing the possible structures of English sentences. The human experts are also required to provide some labeled sentences, where each word is given its actual POS tag. Based on such information, the machine learning researchers inferred the probabilistic distribution of each word belonging to different POS tags, and found the POS tag sequence of a new sentence with highest joint probability that conforms to one of the given grammar trees.

1.1.1 Instance-based feedback

The work on enriched label-related feedback explored in this thesis falls under the umbrella of instance-based methods. However, instance-based methods are still a broad category. A sub-categorization will help to better locate enriched label-related feedback in instance-based methods. Briefly, instance-based methods can be sub-categorized based on the unit where feedback is provided:

- Group-based methods ask the human expert for the aggregated feedback of a group of data instances. In other words, the feedback from the human expert is based on a group of instances, indicating the aggregation of the feedback of all instances in this group. Multiple aggregate functions have been proposed in existing works. Multiple instance learning [Zhou and Zhang, 2002, Settles et al., 2008b] learns a binary classifier from groups of instances that are labeled by human experts as positive, if at least one instance in the group is positive, otherwise, the group is negative. Learning from label proportions [Luo and Hauskrecht, 2018a, Luo and Hauskrecht, 2018b, Luo and Hauskrecht, 2019, Luo and Hauskrecht, 2020] learns a binary classifier from groups of instances that are labeled by humans with proportion estimates, that is probability (or percentage) of positive instances in a group. The main disadvantage of group-based methods is the difficulty in group definition. Clearly the number of possible groups is exponential in the number of instances. Existing methods [Luo and Hauskrecht, 2018a, Luo and Hauskrecht, 2018b] define groups as hyper-cubes aligned to the coordinate axes in the feature space.
- Grouplet-based methods ask the human expert for the ordering or similarity information of a fixed-size small group. Unlike group-based methods, the grouplets in grouplet-based methods are always of a small fixed size (typically no more than four), the human expert provide the ordering or similarity information among the instances in this grouplet rather than aggregated feedback. [Joachims, 2002] for binary classification asked human experts to provide the ordering between two instances, since the positive instances should rank superior to the negative ones. There are also methods asking human experts for similarity information on triplets or quadruplets. Similarity information on triplets includes two instances that

are similar and one instance that is dissimilar to the former two; similarity information on quadruplets includes two instances that are similar and two instances that are dissimilar. Similarity information on triplets or quadruplets was first proposed for the learning of distance metrics [Heim et al., 2015, Heim et al., 2014, Heim and Hauskrecht, 2015], however, [Zhai et al., 2019, Chen et al., 2017] applied such information into multi-class classification since similarity information on triplets or quadruplets help aggregate intra-class instances and dis-separate inter-class instances. The main disadvantage of grouplet-based methods is fallacy hidden in the premise that the ordering or similarity information among instances is easy to obtain. This premise is true in some classification tasks when the orderings or similarities are explicit (e.g. images). However, when the ordering or similarity information is implicit or the evaluation of instances requires professional backgrounds (e.g. diagnosis), obtaining such information can be extraordinarily costly since the human annotator must evaluate all the instances in the grouplet. Another disadvantage of grouplet-based methods is the grouplet number. The maximal number of grouplets is K-polynomial to the instance number (K is the grouplet size), which may limit the scalability of grouplet-based methods.

- Individual-instance-based methods directly ask human experts for feedback on individual data instances. Traditional instance-based methods only ask for a class label, that is, the class an individual instance belongs to. However, obtaining class labels can be very costly: in real-world classification tasks, the average time consumption of obtaining one class label ranges from minutes to hours. More sophisticated instance annotation methods ask for additional information that enhances or elaborates traditional class label feedback with information related to expert's agreement with the label. We refer to such a feedback as enriched label related feedback and it is the main focus of the study in this thesis.

1.1.2 Enriched-label related feedback

Enriched label-related feedback assumes the human expert is able to provide, in addition to the class label, also information on his/her agreement with that label. The premise is simple: an expert who reviews the instance and gives a subjective class label can often provide us with additional information, reflecting the agreement or his/her uncertainty in the label decision. For example,

in binary classification, the human can differentiate examples that clearly, weakly or marginally represent a class. It is this type of information we seek to collect and incorporate into the model building process. Please note, we expect the added cost of this feedback is small compared to the cost of instance review and class-label decision.

There are multiple forms of enriched label-related feedback in different classification scenarios. The first form of enriched label-related feedback, to our knowledge, is probabilistic scores [Nguyen et al., 2011a, Nguyen et al., 2011b, Nguyen et al., 2013] in binary classification scenarios. Basically, in addition to a subjective class label, the human expert is also asked to provide a probabilistic score with additional information, reflecting his/her agreement in the label decision. Another form of enriched label-related feedback in binary classification scenarios is the Likert-scale feedback, where the human expert directly provides the agreement of class labels as ordinal categories. For example, when obtaining a feedback from a physician on whether the patient suffers from a particular disease or not, the physician can also provide his/her agreement in the presence of the disease on a 5-point Likert-scale feedback if he/she agrees, weakly agrees, is neutral, weakly disagrees, or disagrees with the disease. Probabilistic scores are also applicable to multi-class classification scenarios. In multi-class classification tasks, each data instance is associated with one of the multiple class labels. In addition to the class label associated with this instance, the human expert is asked to provide a probabilistic score indicating the agreement to the class label. There are also other forms of enriched label-related feedback reflecting the orderings with other classes/labels. In multi-class classification scenarios, the human expert can also be asked for the alternative classes: if the human expert does not highly agree with the class label of the data instance, s/he may also provide some other classes as alternative choices. A similar form of enriched label-related feedback also exists in multi-label classification scenarios. In multi-label classification tasks, each data instance is associated with a label vector of multiple binary values: if the binary value is positive, the instance is associated with this label, and vice versa. In multi-label classification tasks, the human expert can also be asked for the total orderings of the positive labels, since the human expert may have different agreement on different positive labels of an instance even though they are all marked by the human annotator as positive.

In this thesis, we aim to explore the different forms of enriched label-related feedback above to improve the model quality while not increase the number of labeled instances. A more detailed

introduction of our completed works to handle these enriched label-related feedback can be found in Section 1.4.

1.1.3 Limitations of enriched label-related feedback

Although enriched label-related feedback provides additional information for a data instance obtained from the annotator, it still suffers from the following risks which push us to move forward with it cautiously.

First, some forms of enriched label-related feedback may also contain noise along with additional information. This typically happens when the agreement on the classes/labels are presented as exact values. For example, when the annotator provides the confidence using probabilistic scores, these probabilistic score can be inconsistent and inaccurate [Nguyen et al., 2011a]. If the classification models are overly focused on these exact values, the hidden noise may negatively limit the performance of the classification models.

Second, the cost or time consumption of obtaining enriched label-related feedback may become non-trivial if the enriched label-related feedback is defined improperly. For example, in multi-class classification scenarios, we can ask the annotator to provide the confidence of all classes for each instance. If so, the annotation cost per instance will increase drastically when the class number is large, which contradicts our assumption that enriched label-related feedback can be obtained at a trivial cost and time consumption compared with obtaining the traditional class label of this instance.

Third, the time complexity of the classification models may become intolerable if the enriched label-related feedback is utilized improperly. For example, to eliminate the risk of the noise hidden in probabilistic scores, [Nguyen et al., 2011a, Nguyen et al., 2011b] proposed a method extracting pairwise orderings among instances from the probabilistic scores. However, the number of pairwise orderings is quadratically proportional to the instance number, leading to poor scalability, which limits the deployment of this method on larger datasets.

To eliminate the risks of enriched label-related feedback, we propose different techniques in this thesis. For example, our methods incorporating probabilistic scores are focused on the ordinal categories of confidence rather than the exact values, which eliminate the risks of the noise

hidden in the probabilistic scores. These methods are also focused on the ordinal categories rather than pairwise ordering among data instances, which reduce the time complexity. In the following subsections, we give a general overview of the works related to enriched label-related feedback we have completed, including the idea of utilizing different forms of enriched label-related feedback and how to alleviate the potential risks in different classification scenarios.

1.2 Active learning

Active learning [Lewis and Gale, 1994] is one of the most popular research directions for the problem of optimizing the time and cost of labeling. In active learning, model training and data instance annotation process are interleaved. Briefly, active learning sequentially selects and labels originally unlabeled instances that are most informative and believed to have the most significant potential to improve the model. The main challenge is for this work is to propose an active learning strategy is that it is highly related to the form of the enriched label-related feedback and the classification scenario. To address the problem, we propose complementary active learning strategies for different forms of enriched label-related feedback in different classification scenarios.

There are multiple ways to assess the informativeness of an unlabeled instance. Perhaps the most popular strategy is uncertainty sampling [Lewis and Gale, 1994] which finds the unlabeled instance closest to the decision boundary of the classification model. However, uncertainty sampling is incompatible with enriched label-related feedback, since enriched label-related feedback of an instance indirectly reflects the distance of this instance to the decision boundary. Another popular strategy is query-by-committee [Seung et al., 1992, Tosh and Dasgupta, 2018] that trains a committee of multiple classification models and selects the unlabeled instance on which the committee disagrees the most. The models in the committee can be acquired from different training sets via, for example, bootstrapping all data instances [Breiman, 1996]. The limitation of query-by-committee is a potential bias introduced by the trained models. There are also more sophisticated active learning strategies named expected model change (EMC) which estimates the expected change that the unlabeled instance may bring to the classification model. Briefly, the strategy calculates the change in the model by assuming an unlabeled

instance being assigned to one of the possible labels and weights the change by an estimate of its probabilistic distribution of possible labels. The first implementation of expected model change [Tong and Koller, 2000, Settles et al., 2008b] for binary classification models with merely class labels measures the model change as the change of the model parameters. However, a large change of the parameters does not necessarily imply a large change in the predictions. Therefore, such model change typically overestimates the informativeness of the unlabeled instances. Moreover, the measurement of model change is also highly dependent on the feedback. In this thesis, we propose different approaches to measure different forms of enriched label-related feedback. For Likert-scale feedback, where the agreement of the class label is represented as one of the multiple ordinal categories, the model change can be estimated via the change of all the unlabeled instances on the predicted ordinal category. This is because such change on the ordinal category can reflect the change on the output of the classification model: if the predicted ordinal category does not change, the change on the output of the classification model is also negligible. For probabilistic scores, the model change can be estimated via the change of all the unlabeled instances on the output of the classification model. The model change can be also estimated via the change of all the unlabeled instances on the predicted category if we discretize the range of the probabilistic scores into multiple ordinal categories. For the alternative class choices in multi-class classification scenarios, the model change can be estimated via the change of all the unlabeled instances on the orderings of all the classes. We also emphasize the changes on the highly ranked classes since the change on them may affect the predicted class label. Similarly, for the total orderings of positive classes in multi-label classification scenarios, the model change can be estimated via the change of all the unlabeled instances on the orderings of all the positive labels. Moreover, the change of the binary prediction of each label should also be considered: if the one label changes from positive to negative, or vice versa, this change is greater than the change on orderings and should be emphasized.

In this thesis, we aim to explore the expected model change active learning strategy and tailor such strategy to be compatible with different forms of enriched label-related feedback in different classification scenarios to improve the model quality while not increase the number of labeled instances. A more detailed introduction of our completed works on the expected model change active learning strategy for different forms of enriched label-related feedback can be found in

Section 1.4.

1.3 Main hypothesis

The main hypothesis of our work is that with enriched label-related feedback and corresponding model learning strategies, the classification models can be built more efficiently, that is, with the same number of annotated instances, our methods can build higher performing models than methods that only utilize class-label information. Moreover, we hypothesize that learning from this new type of feedback can be further improved using matching active learning strategies. Please note that the new feedback and active learning are complementary approaches, and hence, can reduce the annotation effort both individually and jointly. In this thesis, we aim to explore the following hypotheses:

H1. *Data with enriched label-related feedback and active learning can reduce the annotation effort both individually and jointly in binary classification scenarios;*

H2. *Data with enriched label-related feedback and active learning can reduce the annotation effort both individually and jointly in multi-class classification scenarios;*

H3. *Data with enriched label-related feedback and active learning can reduce the annotation effort both individually and jointly in multi-label classification scenarios.*

In the following section, we briefly introduce the problems and methods for enriched label-related feedback and active learning methods we have developed in this thesis to study the above hypotheses and provide pointers to the corresponding chapters in the remainder of the document.

1.4 Methods developed in the thesis

1.4.1 Learning of binary classification models from probabilistic scores and Likert-scale feedback

Binary classification scenario is where each data instance belongs to one of the two classes (typically class 0 and class 1). In binary classification scenario, the enriched label-related feedback typically comes in two forms: (1) probabilistic scores, or (2) Likert-scale feedback [Likert, 1932].

Briefly, probabilistic scores are probabilities ranging in $[0, 1]$ (both inclusive), while Likert-scale feedback defines a set of ordinal categories. These two types of information can be used to indicate the strength of agreement (or belief) in the respective class labels. For example, when obtaining feedback from a physician on whether the patient suffers from a particular disease or not, the binary true/false feedback can be refined by obtaining physician's belief in the presence of the disease on a probabilistic score (e.g., 70%), or a 5-point Likert-scale (e.g., strongly agree) by asking if s/he agrees, weakly agrees, is neutral, weakly disagrees, or disagrees.

Existing works [Nguyen et al., 2011a, Nguyen et al., 2011b] convert the probabilistic scores and Likert-scale feedback into pairwise orderings. Such methods show good performance and robustness against the noise in probabilistic scores. However, the number of pairwise orderings is quadratically proportional to the instance number, leading to the poor scalability of these methods. [Nguyen et al., 2013] developed new a new method based on ordinal regression [Chu and Keerthi, 2005] to learn the classification model from such two types of information and demonstrate its benefits over methods based on only class-label information. This method, apart from showing good performance and robustness against the noise in probabilistic scores, is also more scalable since the number of ordering relations obtained from ordinal regression is only linearly proportional to the instance number. However, [Nguyen et al., 2013] left a key quantity, the number of ordinal categories, undetermined. In this thesis, we propose to find the optimal number of ordinal categories via Freedman-Diaconis rule [Freedman and Diaconis, 1981].

To further improve the annotation efficiency of the above methods we enhance them with active learning strategies specifically tailored to the feedback they work with. Briefly, we propose an expected model change (EMC) [Tong and Koller, 2000, Settles et al., 2008b] active learning

strategy for both probabilistic scores and Likert-scale feedback. Such a strategy estimates the expected change of the predictions from all the add-one models of an unlabeled example. An add-one model of the unlabeled example is the model where adding this unlabeled example and a presumed probabilistic score or Likert-scale label into the labeled data. We use such an expected change of the unlabeled examples to select data examples that may help the model the best. To prevent the re-training of add-one models, which is typically inefficient and required for traditional EMC strategy, we also train the add-one models incrementally from the current model, which remarkably reduces the time consumption.

The details of methods for learning binary classification models from probabilistic scores will be discussed in Chapter 3; the details of learning of binary classification models from Likert-scale feedback will be discussed in Chapter 4.

1.4.2 Learning of multi-class classification models from probabilistic scores and ordered class sets

Multi-class classification models are typically learned from annotated data in which every data instance is associated with one class label indicating the top class choice assigned to it from among multiple classes (more than two) by a human annotator. In the multi-class classification scenario, the enriched label-related feedback can come in two forms (1) a probabilistic score, or (2) an ordered class set (OCS).

The probabilistic scores are similar to those used in the binary classification scenario. However, the ordered class sets are much different: human annotators can often express and provide additional information about the top class and its relation to other class choices. For example, when the annotation of a data instance is not a clearcut case, there are other likely class choices the annotator may have in mind. Associating multiple competing classes with one instance is common in various diagnostic tasks. For example, in the medical domain, a list of competing diagnostic classes is referred to as a differential diagnosis. Briefly, given the features (symptoms, observations, etc.) of a patient, the physician considers not only the leading diagnosis (class) but also other alternative diagnoses (classes) that are possible and may fit the patient's case. More specifically, apart from the top class label for each data instance, we let the annotator also provide

information about other alternative classes, and express their descending priorities (or confidence) in the ordered set of classes.

Existing works [Nguyen et al., 2011a, Nguyen et al., 2011b, Nguyen et al., 2013] learning from probabilistic scores are only applicable for binary classification tasks. In this thesis, for the probabilistic scores, we use similar techniques in Section 1.4.1 based on ordinal regression [Chu and Keerthi, 2005]. OCS is a new form of enriched label-related feedback in multi-class classification tasks. To our knowledge, the work in this thesis is the first work utilizing such feedback. We build our methods for utilizing OCS in the learning process by splitting each OCS into two subsets: the subset with higher priority and the one with lower priority. Then we extract ranking information [Joachims, 2002, Herbrich et al., 1999] between these two subsets and incorporate them into approximate multi-class support vector machine (AMSVM) [He et al., 2012]. Our methods, apart from showing a good performance and robustness against the noise in probabilistic scores, is also more scalable since the number of ordering relations obtained is only linearly proportional to the instance number and to the class number.

Similarly to the binary classification problem, we aim to enhance the multi-class classification learning also with active learning methods. For the probabilistic scores, we propose an expected approximate performance change (EAPC) whose inspiration is similar to EMC in Section 1.4.1. For the ordered class sets, we propose a new variant of expected model change (EMC) [Tong and Koller, 2000, Settles et al., 2008b]. Briefly, when adding an unlabeled instance and a possible OCS of this instance into the current model, it calculates the change in the ordering induced by all one-vs-rest classifiers over all unlabeled instances. Since the OCS number of each instance is extremely large (factorial of the class space size), we also propose an approximation that subsamples the OCS's of each unlabeled instance using t-test to find the optimal subsample size. To prevent the re-training of add-one models, which is typically inefficient and required for traditional EMC strategy, we also incorporate multiple techniques that remarkably reduce the time consumption. For EAPC for probabilistic scores, we approximate the projection change of each instance from the corresponding add-one models instead of training them. For EMC for OCS, we train the add-one models incrementally and approximate the OCS distribution by subsampling.

The details of our multi-class classification learning methods with probabilistic scores will be discussed in Chapter 5; the details of multi-class classification learning methods with OCS will be

discussed in Chapter 6.

1.4.3 Learning of multi-label classification from permutation subsets as multi-label ranking

Multi-label classification models are typically learned from annotated data instances where each data instance is associated with a binary label vector, and where each scalar in the binary label vector has a $0/1$ binary value indicating whether the data instance is relevant to this label or not. Human annotators can often provide additional information about the total orderings of the relevant labels (the labels annotated as 1 in the label vector) apart from the label vector itself. For example, when the relevant labels of a data instance are of different certainties, the annotator may provide such information via a permutation subset, which is useful for learning. Such permutation subset of a data instance indicates the annotator's certainties towards the relevant labels in descending order. For example, in an image classification task, the annotator may be certain that there is a house in the image, while s/he may not be so certain whether there is also a dog, because the dog in the image is fuzzy and hard to recognize.

The learning of multi-label classification models from permutation subsets is identical to the learning of multi-label ranking models: multi-label ranking is a learning problem where the goal is to not only identify relevant labels from a set of predefined labels, but also to rank them according to their relevance to a data instance. Consequently, multi-label ranking can be considered as a generalization of multi-label classification and label ranking. Therefore, the key to learning a successful multi-label ranking model is the capture of the dependencies among the labels. However, existing works [Zhou et al., 2014, Jung and Tewari, 2018, Bucak et al., 2009] of multi-label ranking ignore the dependencies among the labels and are focused on the marginal probabilities of the labels. In this thesis, to capture the dependencies among the labels, we propose a multi-label ranking method that combines an auxiliary multi-label ranking support vector machine (MLRSVM) to effectively incorporate such permutation subsets into an existing multi-label classification model to reduce the annotation effort. We also show such auxiliary MLRSVM is a general multi-label ranking model than can be combined with any existing multi-label classification models that support gradient-based learning algorithms. Such a multi-label ranking method can also be applied to the learning of multi-label classification

models from permutation subsets. By conducting experiments on multiple datasets, our method shows it can successfully capture dependencies among labels, leading to better performance when compared with existing methods.

We propose a new active learning strategy based on the expected model change (EMC) [Tong and Koller, 2000, Settles et al., 2008b] also for the permutation subsets in the multi-label classification scenarios. Briefly, when adding an unlabeled instance and a possible permutation subset of this instance into the current model, it calculates the change in the rankings of the relevant labels overall unlabeled instances. The relevant labels are given by the existing multi-label classification model, while the rankings of the relevant labels are given by the auxiliary multi-label ranking model. Since the permutation subset number of each instance is extremely large (exponential of the label space size), we also propose an approximation that subsamples the permutation subsets of each unlabeled instance using t-test to find the optimal subsample size. To prevent the re-training of add-one models, which is typically inefficient and required for traditional EMC strategy, we also train the add-one models incrementally and approximate the permutation subset distribution by subsampling, which remarkably reduces the time consumption.

The details of our ranking-based multi-label classification learning methods with permutation subsets will be discussed in Chapter 7.

2.0 Background

In this chapter, we outline the background and related work for the methods we describe in this thesis. We start with the basics and methods of binary classification learning, along with extension to multi-class and multi-label classification, then discuss two techniques to reduce annotation efforts: active learning and learning with enriched label-related feedback. We use the following notation throughout this thesis: matrices are denoted by capital letters, vectors by boldface lowercase letters and scalars by regular lowercase letters.

2.1 Binary classification learning

Binary learning is a sub-field of machine learning, where the task is to learn a mapping from input examples to desired $0/1$ outputs. In the standard binary classification setting, training data consist of examples and corresponding labels (targets), which are given by a teacher (labeler). The goal is to learn a model that can accurately predict labels of unseen future examples. Formally, given training data $D = \{d_1, d_2, \ldots, d_N\}$ where d_i is a pair of $\langle \mathbf{x}_i, y_i \rangle$, $\mathbf{x}_i$ is an input feature vector, y_i is a desired $0/1$ output given by a teacher, the objective is to learn a mapping function $f : X \rightarrow Y$ such that for a new future example $\mathbf{x}_0$, $f(\mathbf{x}_0) \approx y_0$. Binary classification learning has many applications in practice, for example, given data for past patients, predict whether a new patient has disease or not.

The exact form of the model $f : X \rightarrow Y$, and the algorithms used to learn it, can take on different forms. For example, the model can be based on: logistic regression [McCullagh and Nelder, 1989], a simple (perhaps the most simple) classifier minimizing the generalization error; linear discriminant analysis (LDA) [Fisher, 1936], a mixture-of-Gaussian model performing better on unbalanced data; support vector machine (SVM) [Cortes and Vapnik, 1995, Vapnik, 1995], which formulates an optimization problem with a global optimum, and can also be adapted then applied to high-dimensional data [Hastie et al., 2009, Joachims, 1998]; naive Bayes models [Domingos and Pazzani, 1997], assuming the conditional

independence among the input features; decision trees (classification trees) [Breiman et al., 1984], creating rectangle decision boundaries among the instance; neural networks [Hastie et al., 2009, Van Der Malsburg, 1986, Rumelhart et al., 1986, Cybenko, 1989], creating different layers of neurons to achieve non-linearity, where each neuron is a simple model. In addition, there are various ensemble methods, such as bagging [Breiman et al., 1984], random forests [Ho, 1995, Ho, 1998], boosting [Schapire, 1990], and gradient boosting [Friedman, 2002, Mason et al., 2000], where multiple weak learners are combined to create a strong one.

In this section, we will describe in more detail max-margin models [Cortes and Vapnik, 1995, Vapnik, 1995, Hastie et al., 2009, Joachims, 1998] which is a widely used baseline in classification learning research. Moreover, some of our methods in this thesis are based on max-margin models so their review should help to understand them better.

2.1.1 Max-margin models

The main idea of the max-margin for classification is to find the decision hyperplane that maximizes the margin between instances of the two classes. Here "margin" is defined as the distance from the closest instances to the decision hyperplane. The intuition is that among all possible decisions, the max-margin decision has the best generalization ability. In other words, it has the best chance to classify a future instance correctly. This intuition is proved to be true. In fact, the idea has a strong foundation in statistical learning theory: [Cortes and Vapnik, 1995, Vapnik, 1995] proved that the bound on generalization error is minimized by maximizing the margin.

Figure 1 illustrates this idea. In Figure 1-left positive and negative examples can be perfectly separated by many linear decision boundaries. However, as argued by [Cortes and Vapnik, 1995, Vapnik, 1995], the optimal solution is the decision boundary that maximizes the margin between positive and negative examples (Figure 1-right). Note that the decision hyperplane is determined only by the examples on the margin hyperplanes (circled points in Figure 1-right). Hence, these examples are called "support vectors". In machine learning literature, Max-margin models for classification are often referred by the term "support vector machines (SVM)".

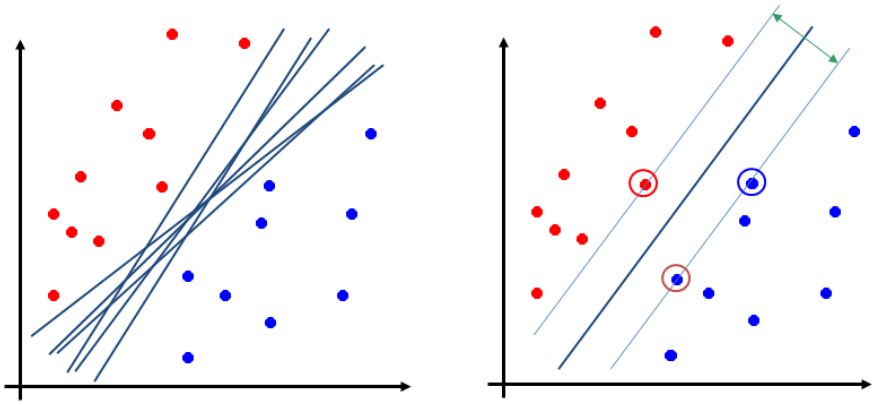

Figure 1: Max-margin (support vector machines) idea. Left: many possible decisions; Right: maximum margin decision.

2.1.2 Linear SVM

Now we will start with a simple case when data are linearly separable. Figure 1 illustrates a 2D example of this case. Linear SVM can be formulated by the following constrained optimization problem:

$$\min_{\mathbf{w},w_0} R(\mathbf{w})$$
$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 \qquad \forall i = 1, 2, \ldots, N$$

where N is the number of examples in the training data, $\mathbf{w}$ is the weight vector of the model to be learned. $\mathbf{w}$ defines the direction of the decision boundary. w_0 is the bias term, which defines the shift of the boundary. $\mathbf{x}_i$ and $y_i \in \{-1, 1\}$ are feature vector and label, respectively, of instance i. $R(\mathbf{w})$ is a regularization function, which is typically written in $L2$ norm in machine learning literature, but in general can be in $L1$ norm. For classification, a new instance $\mathbf{x}$ is assigned 1 (positive) if $\mathbf{w}^T\mathbf{x}_i + w_0 > 0$, otherwise -1 (negative).

The above SVM formulation is called hard-margin SVM, because it requires all instances of the two classes to be linearly separable. However, in practice, it is often impossible to separate

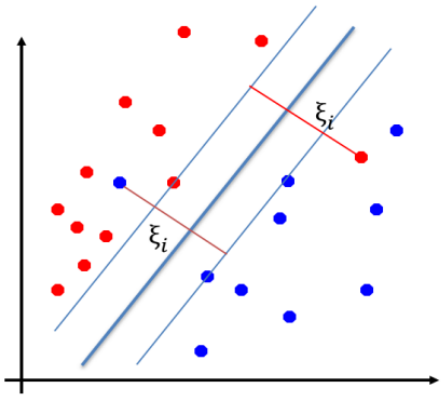

Figure 2: Soft-margin SVM for the linearly non-separable case. Slack variables ξ_i represent distances between $\mathbf{x}_1$ and margin hyperplanes.

data perfectly with a linear boundary, as shown in Figure 2. To handle this case, [Vapnik, 1995] relaxes the above requirement by allowing SVM to make mistakes, but mistakes are penalized in the objective function. We have the following formulation of soft-margin SVM, also called the primal form of soft-margin SVM:

$$\begin{aligned} &\min_{\mathbf{w},w_0} R(\mathbf{w}) + C\sum_{i=1}^{N}\xi_i \\ y_i(\mathbf{w}^T\mathbf{x}_i + w_0) &\geq 1 - \xi_i \qquad \forall i = 1, 2, \ldots, N \\ \xi_i &\geq 0 \end{aligned} \tag{2.1}$$

Slack variables ξ_i represent distances between $\mathbf{x}_i$ and margin hyperplanes. Note that $\xi_i = 0$ if $\mathbf{x}_i$ is located on the correct side of the margins, otherwise $\xi_i > 0$. $\xi_i = \max[0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + w_0)]$ is called the hinge loss. Constant C is a trade-off parameter that defines how much misclassified examples should be penalized. In fact, hard-margin SVM is a special case of soft-margin SVM with C set to positive infinity. Therefore, further in this document, the term "Support Vector Machines" refers to soft-margin SVM.

Both hard and soft-margin formulations are convex optimization problems [Hastie et al., 2009], which means that any local optimum is also the global optimum. This

property is essential because it indicates that if we can find the best local solution, we are guaranteed to have the best global solution. This is not the case for many other classification methods (logistic regression, neural networks, etc.) [Hastie et al., 2009], where we may be "trapped" in local optima and never find the global optimum.

2.1.3 Kernel SVM

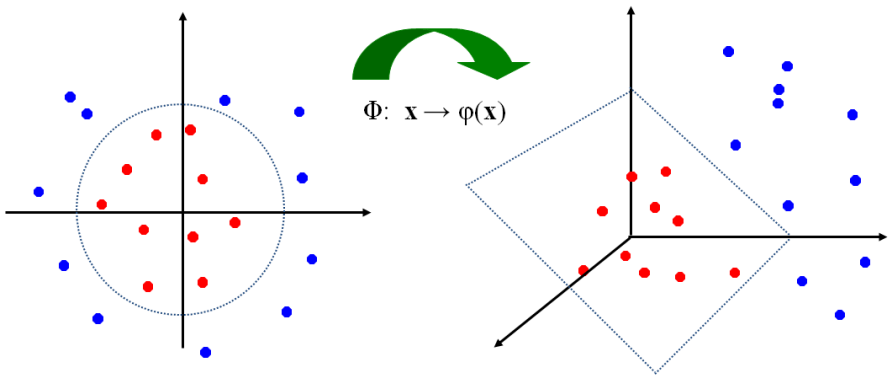

Figure 3: Kernel SVM idea. Left: original 2D input space, positive and negative instances are not linearly separable; Right: function ϕ mapping original input space to a higher-dimensional (3D) feature space, where positive and negatives can be linearly separable.

Linear SVM with soft margins is a powerful classifier when the non-separability is caused by a small number of outliers. However, if data are highly non-linear and are not separable by a linear boundary, e.g., data in Figure 3 (left), then Linear SVM may not perform well. This is often the problem for text and image data [Joachims, 1998]. Kernel SVM [Boser et al., 1992, Theodoridis and Koutroumbas, 2008, Joachims, 1998] was designed to solve this problem. The idea is to map features from the original space to a new higher dimensional space, where linear relations may exist. Figure 3 illustrates this idea: Figure 3-left shows positive and negative examples that cannot be separable in the 2D space; Figure 3-right shows that mapping ϕ of input data from the original 2D space to a 3D space may introduce a linear boundary that can separate examples of two classes (in this case the linear boundary is a surface).

Solving the optimization Equation 2.1 in the feature space is equivalent to solving the

optimization of the following Lagrangian function:

$$\min_{\mathbf{w},w_0,\boldsymbol{\alpha}} L(\mathbf{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||_2^2 - \sum_{i=1}^{N} \alpha_i\{y_i[\mathbf{w}^T\phi(\mathbf{x}_i) + w_0] - 1\}$$

where $\boldsymbol{\alpha} = \{\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_N\}$ is the vector of Lagrangian multipliers. Note that for the demonstration purpose we use $L2$ norm regularization $||\mathbf{w}||_2$, which is widely used in the machine learning literature.

Setting the derivatives of $L(\mathbf{w},\ w_0,\ \boldsymbol{\alpha})$ with respect to $\mathbf{w}$ and w_0 equal to 0, we obtain the following two conditions:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \phi(\mathbf{x}_i)$$
$$0 = \sum_{i=1}^{N} \alpha_i y_i$$

Plugging these conditions into $L(\mathbf{w}, w_0, \boldsymbol{\alpha})$ gives the dual form of the max-margin model:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$
$$0 \geq \alpha_i \geq C \qquad \forall i = 1, 2, \ldots, N$$
$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

where $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ is a kernel function. For Linear SVM, $K(\mathbf{x}_i, \mathbf{x}_j)$ is the dot product of $\mathbf{x}_i$ and $\mathbf{x}_j$: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$.

Solving constrained optimization problems in high dimensional spaces is difficult and computationally expensive [Boser et al., 1992]. Therefore, kernel functions $K(\cdot, \cdot)$ should be designed so that SVM: (1) has the representation power of high dimensional spaces and (2) still be computationally efficient. This can be done by choosing a mapping from the input space X to a new feature space $F : \mathbf{x} \rightarrow \phi(\mathbf{x})$, such that $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ where $\mathbf{x}_i, \mathbf{x}_j \in X$. Thus, we implicitly compute dot product in a high dimensional space F, in terms of operations in the original low dimensional space X. This is called the "kernel trick".

Many different types of kernels have been designed by the research community. For example, the two most widely used kernels are: (1) polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (c + \mathbf{x}_i \cdot \mathbf{x}_j)^p$ where $p \in Z^+$ is the order and $c \geq 0$ is a constant; (2) radial basis functions (RBF) kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2})$ where $\sigma > 0$ is the standard deviation.

2.1.4 Summary

In this section, we gave a brief review of support vector machines (SVM). More details about theory and analysis of SVM can be found in [Cortes and Vapnik, 1995, Vapnik, 1995, Hastie et al., 2009, Joachims, 1998].

2.2 Multi-class classification learning

In multi-class classification models, "multi-class" indicates that the number of the classes is always greater than 2. Typically, these classes are treated equally, in other words, there is no relationship of orderings or similarities among these classes. For example, for a three-class classification model, the class labels may be represented as class 0, class 1 and class 2. Such representation of class labels may mistakenly indicate to some people that class 2 is closer to class 1 than to class 0. However, the fact is that these three classes are dissimilar with each other without any orderings. In the standard setting of multi-class classification, training data consist of examples and corresponding labels (targets), which are given by a teacher (labeler). The goal is to learn a model that can accurately predict labels of unseen future examples. Formally, given training data $D = \{d_1, d_2, \ldots, d_N\}$ where d_i is a pair of $\langle \mathbf{x}_i, y_i \rangle$, $\mathbf{x}_i$ is an input feature vector, y_i is a desired categorical output given by a teacher, the objective is to learn a mapping function $f : X \to Y$ such that for a new future example $\mathbf{x}_0$, $f(\mathbf{x}_0) \approx y_0$. Multi-class classification learning is also useful in practice, for example, given historical clinical data, predict which exact disease a (future) patient may have.

The exact form of the model $f : X \to Y$, and the algorithms used to learn it can be extended from the binary classification models in Section 2.1. Some methods

for binary classification can be easily extended. For example, naive Bayes models [Domingos and Pazzani, 1997] and decision trees (classification trees) [Breiman et al., 1984] supports multi-class classification without modification, neural networks [Hastie et al., 2009, Van Der Malsburg, 1986, Rumelhart et al., 1986, Cybenko, 1989] only need to modify the output layer. In this section, we will describe multi-class support vector machine (MCSVM) [Vapnik, 1998, Weston et al., 1999] and approximate multi-class support vector machine (AMSVM) [He et al., 2012], which are two popular multi-class extensions of SVM discussed in Section 2.1.1, in more details. Briefly, these two multi-class extensions decompose the multi-class classification task into multiple binary classification tasks, and apply a binary SVM [Cortes and Vapnik, 1995, Vapnik, 1995] for each task. Also, the kernel trick for binary SVM [Hastie et al., 2009, Joachims, 1998] is compatible with these two multi-class extensions. We note, that some of our new methods presented later in the thesis are based on these methods, so a review of them should help one to understand better the following chapters of the thesis.

2.2.1 Multi-class support vector machine (MCSVM)

Our goal is to learn a multi-class classifier $f : X \to Y$, where X is the feature space and $Y \in \{1, 2, \ldots, k\}$ represents class labels of a data instance. Hence each labeled data entry D_i consists of two components: $D_i = \langle \mathbf{x}_i, y_i \rangle$, an input and a class label.

In multi-class support vector machine (MCSVM), we learn k binary support vector machine jointly, one for each class. Briefly, MCSVM works by trying to assure for every training data instance the projection of its assigned class label to be higher than the projection of any other class. Therefore, $(k-1)$ constraints are derived for each labeled data instance, one for each class, except for the assigned class label. The total number of constraints in MCSVM is thus $O(kN)$, where N is the number of labeled data instances. For each data instance, the projection from the binary classifier of the class label should be higher than the projection from other classes. Formally, we would like to get k projection mappings $f_1(\cdot), f_2(\cdot), \ldots, f_k(\cdot)$, such that for each data instance $\mathbf{x}_i$, the projection $f_{y_i}(\mathbf{x}_i)$ is greater than $f_l(\mathbf{x}_i)$ for $l \in \{1, 2, \ldots, k\} \setminus y_i$. To permit some flexibility, we allow violations of the constraints but penalize them through the loss function. Therefore, the multi-class support vector machine is formulated as follows:

$$\min_{W,\boldsymbol{\xi}} \frac{1}{2}\sum_{l=1}^{k}||\mathbf{w}_l||_2^2 + C\sum_{i=1}^{N}\sum_{j\neq y_i}\xi_{i,j}$$
$$(\mathbf{w}_{y_i} - \mathbf{w}_j)^T\phi(\mathbf{x}_i) \geq 1 - \xi_{i,j} \qquad \forall i = 1, 2, \ldots, N \quad \forall j \neq y_i$$
$$\xi_{i,j} \geq 0 \qquad \forall i = 1, 2, \ldots, N \quad \forall j \neq y_i \tag{2.2}$$

where y_i is the class label of $\mathbf{x}_i$ and $\phi(\cdot)$ is the projection of kernel space. $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ are parameters of the k binary one-vs-rest classifiers. N is the number of labeled instances. $\Xi = \{\xi_1, \xi_2, \ldots, \xi_N\}$ are the slack variables for each constraint. For prediction, the class with the highest projection value is selected as the predicted class.

2.2.2 Approximate multi-class support vector machine (AMSVM)

The approximate multi-class SVM (AMSVM) is an approximation of the standard multi-class SVM (MCSVM) method in Section 2.2.1. In AMSVM the set of the constraints is merged and replaced with one constraint that assumes that for each data instance the projection of the class label is higher than the average projection for all the other classes. Via such averaging, the number of constraints is significantly reduced: only one constraint is derived for each labeled data instance. Therefore, the total number of constraints in AMSVM is reduced to $O(N)$. Formally, in the AMSVM with k classes, k binary SVMs $f_1(\cdot), f_2(\cdot), \ldots, f_k(\cdot)$ are trained jointly. For every labeled instance $\langle \mathbf{x}_i, y_i \rangle$, we try to assure the projection $f_{y_i}(\mathbf{x}_i)$ of the class label y_i should be greater than the average projection $\frac{1}{k-1}\sum_{l\neq y_i} f_l(\mathbf{x}_i)$ of all the other classes $l \in \{1, 2, \ldots, k\} \setminus y_i$. The optimization of AMSVM can be formalized as:

$$\min_{W,\Xi} \frac{1}{2}\sum_{l=1}^{k}||\mathbf{w}_l||_2^2 + C\sum_{i=1}^{N}\xi_i$$
$$(\mathbf{w}_{y_i} - \frac{1}{k-1}\sum_{j\neq y_i}\mathbf{w}_j)^T\phi(\mathbf{x}_i) \geq 1 - \xi_i \qquad \forall i = 1, 2, \ldots, N$$
$$\xi_i \geq 0 \qquad \forall i = 1, 2, \ldots, N \tag{2.3}$$

where y_i is the class label of $\mathbf{x}_i$ and $\phi(\cdot)$ is the projection of kernel space. $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ are parameters of the k binary one-vs-rest classifiers. N is the number of labeled instances. $\Xi = \{\xi_1, \xi_2, \ldots, \xi_N\}$ are the slack variables for each constraint. For prediction, the class with the highest projection value is selected as the predicted class. As shown in [He et al., 2012] the performance of AMSVM is often comparable to the standard multi-class SVM (MCSVM).

2.2.3 Summary

In this section, we gave a brief review of two popular multi-class extensions of support vector machine: multi-class support vector machine (MCSVM) and approximate support vector machine (AMSVM). More details about theory and analysis of these two multi-class extensions can be found in [Vapnik, 1998, Weston et al., 1999] and [He et al., 2012] respectively.

2.3 Multi-label classification learning

In multi-label classification models, "multi-label" indicates that the number of the labels is always greater than or equal to 2. In the standard setting of multi-label classification, training data consist of data examples, and each example corresponds to a label vector of multiple binary labels (targets), which are given by a teacher (labeler). The goal is to learn a model that can accurately predict all the binary labels in the label vector of unseen future examples. Formally, given training data $D = \{d^{(1)}, d^{(2)}, \ldots, d^{(N)}\}$ where $d^{(i)}$ is a pair of $\langle \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \rangle$, $\mathbf{x}^{(i)}$ is an input feature vector, $\mathbf{y}^{(i)}$ is a desired label (output) vector of binary values given by a teacher (annotator), the objective is to learn a mapping function $f : X \rightarrow Y$, where X is the feature space and $Y = \{0, 1\}^k$ represents label vector space of a data instance, such that for a new future example $\mathbf{x}^{(0)}$, $f(\mathbf{x}^{(0)}) \approx \mathbf{y}^{(0)}$. Multi-label classification learning is also useful in practice, for example, given historical clinical data, predict all the diseases that a (future) patient may have.

Multi-label classification can be treated as an aggregation of multiple binary classification tasks with the same input feature vector for each data example. Multi-label classification can also be treated as an extension of multi-class classification: in multi-class classification, each instance

is associated with one single category out of all the categorical values; in multi-label classification, each instance can be associated with any number of all the categorical values.

Again, the key to learning a multi-label classification model is the successful capture of the hidden dependencies among the labels. Such hidden dependencies include the fact that some labels may typically coexist. For example, an image with beach is often with ocean as well. Such hidden dependencies also include the fact that some labels may typically be mutually exclusive. For example, an image with beach is rarely with electronics. A successful capture of the hidden dependencies helps the learning of the coexistence and mutual exclusion of the labels and can substantially improve the performance of the multi-label classification model.

The exact form of the model $f : X \rightarrow Y$, and the algorithms used to learn it, can be directly extended from the binary classification models in Section 2.1, or the multi-class classification models in Section 2.2, or probabilistic graphical models based on directed acyclic graphs (DAGs) or undirected graphs. In this section, we will describe binary relevance (BR) [Boutell et al., 2004, Clare and King, 2001], which is directly extended from the binary classification models; labeling powerset (LP) [Tsoumakas et al., 2010], which is directly extended from the multi-class classification models; conditional random field (CRF) [Lafferty et al., 2001, Bradley and Guestrin, 2010, Naeini et al., 2015], which is extended from probabilistic graphical models based on undirected graphs (a.k.a. undirected graphical models, or UGMs); classifier chains [Read et al., 2009] and conditional tree-structured Bayesian network (CTBN) [Batal et al., 2013, Hong et al., 2014, Hong et al., 2015], which are extended from probabilistic graphical models based on DAGs (a.k.a directed graphical models, or DGMs). Although we are not deriving new methods based on these methods, some of our methods mentioned in this thesis can be combined with these methods, therefore it would be useful to have a brief introduction to them.

2.3.1 Binary relevance (BR)

Our goal is to learn a multi-label classifier $f : X \rightarrow Y$, where X is the feature space and $Y = \{0, 1\}^k$ represents label vector space of a data instance. Hence each labeled data entry $D^{(i)}$ consists of two components: $D^{(i)} = \langle \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \rangle$, an input vector and a label vector.

Binary relevance (BR) [Boutell et al., 2004, Clare and King, 2001] is a simple (perhaps the most simple) multi-label classification model that learns k binary classifiers independently. More formally, we would like to get k binary classifiers $f_1(\cdot), f_2(\cdot), \ldots, f_k(\cdot)$ such that for new future instance $\mathbf{x}^{(0)}$, $f_j(\mathbf{x}^{(0)}) \approx y_j^{(0)}$ for any $j \in \{1, 2, \ldots, k\}$. Such k binary classifiers $f_1(\cdot), f_2(\cdot), \ldots, f_k(\cdot)$ are trained independently, in other words, for any $j \in \{1, 2, \ldots, k\}$, $f_j(\cdot)$ is trained only on $\mathbf{x}^{(i)}$ and $y_j^{(i)}$ for $i \in \{1, 2, \ldots, N\}$.

Clearly, the limitation of binary relevance is that it totally ignores the hidden dependencies among the labels, which is the key to learning a well-performed multi-label classification model.

2.3.2 Labeling powerset (LP)

Labeling powerset (LP) [Tsoumakas et al., 2010] is another simple multi-label classification model that learns the powerset of k labels. More formally, there are 2^k different outcomes for a label vector $\mathbf{y}$ in the label vector space $Y = \{0, 1\}^k$ with k labels. Labeling powerset multi-label classification model first constructs a one-to-one mapping $g : Y = \{0, 1\}^k \to Z = \{1, 2, \ldots, 2^k\}$ to map each outcome of a label vector into a categorical value, then learns a multi-class classifier $f : X \to Z$ such that for a new future instance $\mathbf{x}^{(0)}$, $f(\mathbf{x}^{(0)}) \approx g(\mathbf{y}^{(0)})$.

Labeling powerset successfully captures the hidden dependencies among the labels by learning the full-joint of the labels. However, the limitation is also obvious: the number of the categorical values is exponential to the number of labels, which limits the scalability of this method. Also, this method cannot learn the outcomes that are absent in the label vectors of the training data.

2.3.3 Classifier chain (CC)

Classifier chain (CC) [Read et al., 2009] is a directed graphical model. Briefly, CC learns a linear chain to model the conditional likelihood over all the labels, where each label is dependent on all its former labels on the chain. More formally, we would like to obtain a decomposition of the likelihood $P(\mathbf{y}|\mathbf{x}) = \prod_j P(y_j|\mathbf{x}, \pi(y_j))$, where $\pi(y_j)$ includes all the former labels of label y_j in the linear chain, such that for new future instance $\mathbf{x}^{(0)}$, $P(\mathbf{y}^{(0)}|\mathbf{x}^{(0)}) > P(\mathbf{y}|\mathbf{x}^{(0)})$ for any $\mathbf{y} \neq \mathbf{y}^{(0)}$.

2.3.4 Conditional random field (CRF)

Conditional random field (CRF) [Lafferty et al., 2001, Bradley and Guestrin, 2010, Naeini et al., 2015] is an undirected graphical model. Briefly, CRF learns an undirected graph to model the pairwise dependencies between each pair of labels. Such undirected graph can be a tree [Lafferty et al., 2001, Bradley and Guestrin, 2010] where the prediction of each instance can be calculated in linear time complexity, or an arbitrary undirected graph requiring approximation to reduce the time complexity for prediction [Naeini et al., 2015]. More formally, we would like to obtain a projection mapping $f(\mathbf{x}, \mathbf{y}) = \prod_{j,l} \psi(\mathbf{x}, y_j, y_l)\phi(\mathbf{x}, y_j)$, where $\psi(\mathbf{x}, y_j, y_l)$ is the pairwise potential function for label y_j, y_l, and $\phi(\mathbf{x}, y_j)$ is the individual potential function for label y_j, such that for new future instance $\mathbf{x}^{(0)}$, $f(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}) > f(\mathbf{x}^{(0)}, \mathbf{y})$ for any $\mathbf{y} \neq \mathbf{y}^{(0)}$.

2.3.5 Conditional tree-structured Bayesian network (CTBN)

Conditional tree-structured Bayesian network (CTBN) [Batal et al., 2013, Hong et al., 2014, Hong et al., 2015] is a directed graphical model. Briefly, CTBN learns a directed tree [Batal et al., 2013] to model the conditional likelihood over all the labels. More formally, we would like to obtain a decomposition of the likelihood $P(\mathbf{y}|\mathbf{x}) = \prod_j P(y_j|\mathbf{x}, \pi(y_j))$, where $\pi(y_j)$ is the parents of label y_j in the directed tree, such that for new future instance $\mathbf{x}^{(0)}$, $P(\mathbf{y}^{(0)}|\mathbf{x}^{(0)}) > P(\mathbf{y}|\mathbf{x}^{(0)})$ for any $\mathbf{y} \neq \mathbf{y}^{(0)}$. CTBN can also be combined with ensembling methods [Hong et al., 2014, Hong et al., 2015] which provides better performance on predictions.

By modeling the conditional dependencies via undirected or directed networks, multi-label classification models extended from probabilistic graphical models can efficiently capture the hidden dependencies among labels and train the models in polynomial time. Because of that, multi-label classification models extended from probabilistic graphical models are gaining more and more popularity in recent years.

2.3.6 Summary

In this section, we gave a brief introduction of four multi-label classification models: binary relevance (BR) [Boutell et al., 2004, Clare and King, 2001],

labeling powerset (LP) [Tsoumakas et al., 2010], conditional random field (CRF) [Lafferty et al., 2001, Bradley and Guestrin, 2010, Naeini et al., 2015], classifier chain (CC) [Read et al., 2009], and conditional tree-structured Bayesian network (CTBN) [Batal et al., 2013, Hong et al., 2014, Hong et al., 2015]. More details about theory and analysis of these multi-label classification models can be found in the referred papers of each model.

2.4 Learning to rank

Learning to rank [Liu, 2009, Mohri et al., 2012] is a sub-field of machine learning focused on the construction of ranking models for information retrieval or machine learning systems. The training data of ranking models typically consist of instances with some partial or total ordering information specified on the data instances or labels. Such ordering information is typically induced by giving a numerical or ordinal score for each data instance or label. The ranking model aims to rank the future data instances or labels in a similar way to the rankings in the training data.

Regarding the type of ordering information provided by the teacher (labeler), the ranking models can be categorized into three sub-categories: instance ranking [Joachims, 2002, Radlinski and Joachims, 2005], label ranking [Vembu and Gärtner, 2011, Zhou et al., 2014], and multi-label ranking [Zhou et al., 2014, Jung and Tewari, 2018, Bucak et al., 2009]. In this section, we will give a brief introduction to these three sub-categories.

2.4.1 Instance ranking

In the standard setting of instance ranking models [Joachims, 2002, Radlinski and Joachims, 2005], training data consist of examples and some partial or total ordering information specified on the data examples, which are given by a teacher (labeler). The goal is to learn a model that can accurately order the unseen future examples. Formally, given training data $D = \{X_t, S_t\}$, where $X_t = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is the set of instances and $S_t \subset \bigcup_{Z \in \mathcal{P}(X_t)} \mathfrak{G}_A(Z)$ is the set of partial ordering information on X_t, where $\mathcal{P}(\cdot)$ denotes the powerset and $\mathfrak{G}_A(\cdot)$ denotes the automorphism group. The objective is to learn a mapping function

$f : X \to \mathbb{R}$ such that for new future examples $\mathbf{x}_a$ and $\mathbf{x}_b$, the comparison between $f(\mathbf{x}_a)$ and $f(\mathbf{x}_b)$ follows the partial ordering information regarding these two future examples.

2.4.2 Label ranking

In the standard setting of label ranking models [Vembu and Gärtner, 2011, Zhou et al., 2014], training data consist of examples and some partial or total ordering information specified on the labels of each example, which are given by a teacher (labeler). The goal is to learn a model that can accurately order all the labels of the unseen future examples. Formally, given training data $D = \{d_1, d_2, \ldots, d_N\}$, where $d_i = \langle \mathbf{x}_i, S_i \rangle$ is a pair. $\mathbf{x}_i$ is feature vector of the instance and $S_i \subset \bigcup_{Z \in \mathcal{P}(Y)} \mathfrak{G}_A(Z)$ is the set of partial ordering information on the label space $Y = \{1, 2, \ldots, K\}$, where $\mathcal{P}(\cdot)$ denotes the powerset and $\mathfrak{G}_A(\cdot)$ denotes the automorphism group. The objective is to learn a mapping function $f : X \times Y \to \mathbb{R}$ such that for a new future example $\mathbf{x}_0$, the comparison between $f(\mathbf{x}_0, y_j)$ and $f(\mathbf{x}_0, y_l)$ follows the partial ordering information regarding these label j and label l of this example.

2.4.3 Multi-label ranking

Again, multi-label ranking [Zhou et al., 2014, Jung and Tewari, 2018, Bucak et al., 2009] is a learning problem where the goal is to not only identify relevant labels from a set of predefined labels, but also to rank them according to their relevance to a data instance [Zhou et al., 2014]. Consequently, multi-label ranking can be considered as a generalization of multi-label classification and label ranking. In the standard setting of multi-label ranking models, training data consist of examples and the total ordering information specified on all the relevant labels of each example, which are given by a teacher (labeler). The goal is to learn a model that can accurately find the relevant labels and order all the relevant labels of the unseen future examples. Formally, given training data $D = \{d_1, d_2, \ldots, d_N\}$, where $d_i = \langle \mathbf{x}_i, S_i \rangle$ is a pair. $\mathbf{x}_i$ is feature vector of the instance and $S_i \in \bigcup_{Z \in \mathcal{P}(Y)} \mathfrak{G}_A(Z)$ is the total ordering information on the relevant labels Z over the label space $Y = \{1, 2, \ldots, K\}$, where $\mathcal{P}(\cdot)$ denotes the powerset and $\mathfrak{G}_A(\cdot)$ denotes the automorphism group. Typically, the objective is to learn a mapping function $f : X \times Y \to \mathbb{R}$ such that, for a new future example $\mathbf{x}_0$: (1) the comparison between $f(\mathbf{x}_0, y_j)$

and $f(\mathbf{x}_0, y_l)$ follows the total ordering information regarding these label j and label l of this example if label j and label l are relevant labels; (2) $f(\mathbf{x}_0, y_j) > 0$ should hold regarding label j of this example if label j is a relevant label; (3) $f(\mathbf{x}_0, y_l) < 0$ should hold regarding label l of this example if label l is an irrelevant label. Overall, compared with label ranking we reviewed in 2.4.2, multi-label ranking only enforces the orderings among the relevant labels.

In this thesis, we propose new multi-label classification models with permutation subsets. We start by first defining and formalizing the problem of learning from permutation subsets in multi-label settings. Then, we point out that such multi-label classification models with permutation subsets is identical to multi-label ranking models. After that, we present an two-state algorithm for learning the multi-label ranking model. The details of multi-label classification with permutation subsets as multi-label ranking will be discussed in Chapter 7.

2.4.4 Summary

In this section, we gave a brief introduction of the three sub-categories of ranking models: instance ranking [Joachims, 2002, Radlinski and Joachims, 2005], label ranking [Vembu and Gärtner, 2011, Zhou et al., 2014], and multi-label ranking [Zhou et al., 2014, Jung and Tewari, 2018, Bucak et al., 2009]. More details about theory and analysis of these multi-label ranking models can be found in the referred papers of each model.

2.5 Reducing labeling efforts

By definition, supervised learning models rely on labels given in the training data, and in practice, they often must be trained on a large number of labeled examples in order to perform well. However, as mentioned in the Introduction, the process of labeling examples using subjective human assessments faces one severe problem: it can be extremely time-consuming and costly, which results in a limited number of labeled examples. Since supervised learning methods rely on labeled examples, we need to find approaches to obtain more useful information (labels) with lower cost and utilize them efficiently. Again, in this thesis, we focus on classification learning,

where our goal is to build classification models that can learn with smaller training data and make a more accurate prediction on future unseen instances.

In this section, we give an overview of research works that are relevant to our solutions for the above problems. First, we review active learning, which is a sub-field of machine learning that aims to reduce labeling cost by selecting the most informative examples. Then we give an overview of learning with enriched label-related feedback and its relevant research.

2.5.1 Active learning

Active Learning [Lewis and Gale, 1994] is a sub-field of machine learning, where the primary goal is to reduce the cost of labeling examples. Active learning has been explored extensively by the data mining and machine learning communities in recent years. In traditional "passive" learning, the learner randomly picks examples from the database and requests labels for them. However, in active learning, model training and data instance annotation process are interleaved. Active learning sequentially selects and labels originally unlabeled instances that are most informative and believed to have the greatest potential to improve the model. Such potential is also called as the "informativeness" of an unlabeled instance.

There are multiple ways to assess the "informativeness" of an unlabeled instance [Settles et al., 2008b]. Now we will summarize the most popular ones in this section.

2.5.1.1 Uncertainty sampling One popular (perhaps the most popular) strategy is *uncertainty sampling* [Lewis and Gale, 1994]. The core idea of uncertain sampling is the selection of unlabeled instances with the highest uncertainties. Here, high uncertainty indicates that the prediction of an unlabeled instance is prone to change because the learning model fails to provide the prediction of this unlabeled data with high confidence.

Uncertainty sampling has been widely combined with many classification scenarios. For example, in binary classification problems, there are only two classes: class 0 and class 1. If the probabilistic predictions of class 0 and class 1 for an unlabeled instance are similar (both close to 0.5), this unlabeled sample should be considered as uncertain. In multi-class classification scenarios, three different standards are applied to measure uncertainty:

(1) lowest confidence [Lewis and Gale, 1994, Settles et al., 2008b, Culotta and McCallum, 2005], that queries the unlabeled instance with lowest maximum in predictions over all classes, and (2) marginal confidence [Scheffer et al., 2001], that queries the unlabeled instance with lowest discrepancy in its top two class predictions, and (3) information entropy [Settles et al., 2008b, Hwa, 2001], that queries the unlabeled instance with highest information entropy [Shannon, 1948] over predictions of all classes.

The main limitation of uncertainty sampling is that it typically becomes futile when combined with enriched label-related feedback since such enriched label-related feedback implicitly provides the uncertainty of a data instance. Therefore, none of our works on enriched label-related feedback is combined with uncertainty sampling.

2.5.1.2 Query-by-committee (QBC) Another popular strategy is *query-by-committee* (QBC) [Seung et al., 1992] that trains a committee of models and selects the unlabeled instance on which the models disagree the most. Query-by-committee is inspired by the thesis of version space [Mitchell, 1979]. Version space indicates all the learning models that provide the best performance of the training data. A previous active learning strategy using version space is proposed by [Cohn et al., 1996]. When an unlabeled instance comes, the learning models in the version space will provide different predictions. If the predictions vary a lot, that is, the disagreement among the predictions is high, it is better to ask the human labeler to annotate this unlabeled sample to provide a certain label to eliminate such disagreement.

The active learning strategy above suffers from a practical concern that it is usually unfeasible to enumerate all the learning models in the version space. [Haussler, 1989] shows that as the size of version space can be exponential to the size of the training data. To solve this problem, [Seung et al., 1992] proposed query by committee strategy. In query by committee, the version space is substituted by a committee with multiple learning models. These learning models are all trained over the training data, but with different configurations. When an unlabeled instance comes, the learning models in the committee will provide different predictions. If the predictions vary a lot, that is, the disagreement among the predictions is high, it is better to ask the human labeler to annotate this unlabeled sample to provide a certain label to eliminate such disagreement. Apparently, query by committee is quite similar to the previous active learning strategy. In fact,

the committee acts as an approximation of the version space with only limited numbers of learning models [Burbidge et al., 2007].

The main limitation of QBC is a potential bias the trained models introduced by bootstrapping the dataset. Therefore, none of our works on enriched label-related feedback is combined with QBC.

2.5.1.3 Expectation-based strategies Another family of more sophisticated querying strategies is expectation-based strategies. Briefly, the expectation-based strategy calculates the change in the model due to an unlabeled instance being assigned to one of the possible labels, and weights the change by an estimate of its probability. The first expectation-based querying strategy is *expected model change* [Tong and Koller, 2000, Settles et al., 2008b]. The model change is measured regarding the change of the model parameters. However, a big change of the parameters does not necessarily imply a big change in the models predictions. Therefore, this strategy typically overestimates the "informativeness" of each unlabeled instance. Other representative strategies are *expected error reduction* [Roy and McCallum, 2001] and *variance reduction* [Geman et al., 1992]. The first one seeks an example that would let it reduce the generalization error of the model. The second one seeks an example that would minimize the prediction variance of the current model the most.

The main advantage of expectation-based strategies is "knowledgeability": by considering an unlabeled instance as labeled, expectation-based strategies can infer how the model may change in the future. Neither uncertainty sampling nor QBC can achieve this. However, the main disadvantage of expectation-based strategies is low efficiency: typically we have to consider each unlabeled instance to be labeled and each possible label this unlabeled instance may have. What makes things worse for models with enriched label-related feedback, the number of possible labels for each unlabeled instance is typically large. Therefore, reasonable approximation techniques are also preferred for expectation-based strategies.

In this thesis, we propose new expectation-based active learning strategies per form of enriched label-related feedback. We also propose multiple techniques to reduce time consumption. The details of these expectation-based strategies will be discussed in Chapter 3 to 7.

2.5.1.4 Active group learning (AGL) Recent active learning work focuses on more sophisticated querying strategies that go beyond standard instance-based label-oriented queries. *Active group learning* (AGL) [Luo and Hauskrecht, 2018a, Luo and Hauskrecht, 2018b] is one approach that is gaining popularity. Instead of the instance-based labels, AGL constructs queries for subpopulations (groups of examples) the human annotator labels with class proportions. Briefly, AGL is based on the following assumptions: (1) large groups should be split first as they represent a broader input feature space. (2) impure groups (regarding class proportions) should be prioritized as well. (3) the refinement of the above two types groups offers more labeling information, and thus they give rise to faster model change rate and model convergence.

The advantage of the approach is that multiple instances are labeled jointly with just one query. However, AGL is incompatible with enriched label-related feedback, since it can only construct queries for instance groups, yet our classification models with enriched label-related feedback provide feedback for individual instances. Therefore, none of our works on enriched label-related feedback is combined with AGL.

2.5.2 Learning with enriched label-related feedback

Learning with enriched label-related feedback is a relatively new approach for improving the classification learning process. In general, enriched label-related feedback covers additional information provided by a human annotator related to the class/label choice. The idea of learning with enriched label-related feedback is based on a simple premise: enriched label-related feedback can often be provided by human annotators at an insignificant cost when compared to the cost of instance review and label assessment.

In this section, we will give a brief review of different forms of enriched label-related feedback: probabilistic scores, Likert-scale feedback and ordered class sets.

2.5.2.1 Probabilistic scores Perhaps the most intuitive form is a probabilistic score, which has been explored in the context of binary classification problems. Generally speaking, probabilistic scores reflect the different degrees of certainty in binary labels. If a surgeon determines the probability of the emergence of some disease is 70%, the probabilistic score of this event is 0.7.

Such probabilistic scores provide much more refined information regarding the confidence/belief of the class label compared with merely class label. For example, in a raining prediction task, if the output is just a binary label indicating there will be rain, people may still hesitate whether to bring umbrellas or not. However, if the output indicates the precipitation rate is 90%, people may feel more assured that there will certainly be a rain, and they may determine to bring umbrellas; if the output indicates the precipitation rate is only 60%, some people may take the risk rather than bring umbrellas.

It seems that probabilistic scores are easy to incorporate for learning: they look straightforward and precise. On the contrary, probabilistic scores are often corrupted by noise. That is, evaluations from human annotators are often mixed with their subjective prejudice which varies by time. For instance, an annotator in a delighted mood may label with better scores. An annotator may turn upset by previous labels, thus changing criteria of current annotation mission. Such issue has been well documented in literature [Juslin et al., 1998, Griffin and Tversky, 1992].

To learn a robust binary classification model with probabilistic scores, [Nguyen et al., 2011a, Nguyen et al., 2011b, Nguyen et al., 2013] developed a method that focuses on the pairwise orderings among all data examples. Basically, this method learns a parametric discriminative model by attempting to satisfy pairwise score orderings among all data examples while ignoring their exact probabilistic scores. The limitation of the approach is that the number of constraints the orderings induce is quadratic in the number of data examples. Another work by [Thiel, 2008] explored a framework where probabilistic scores are derived from multiple annotators with potential disagreements. This work, however, does not consider any noise. More recently, [Peng and Wong, 2014, Peng et al., 2014] proposed a new non-parametric algorithm for predicting the probability associated with binary classes based on the Gaussian process regression. The method defines the mean function of the Gaussian process to be 0.5 and the covariance function using the Radial basis kernel. The model lets one to predict the probability p_i for any new point x_i by calculating the posterior distribution of the Gaussian process. The limitations of the approach are the design of the covariance function (restricted to the radial basis functions), and the non-parametric nature of the model when it is applied to prediction tasks.

In this thesis, we propose new binary and multi-class classification models with a probabilistic score. Our new classification models can utilize probabilistic scores effectively: it significantly

reduces the annotation effort. Our models are also robust against the noise in probabilistic scores and efficient: by the number of constraints introduced by probabilistic scores is only linear in the number of data instances, which is a significant improvement compared with [Nguyen et al., 2011a, Nguyen et al., 2011b, Nguyen et al., 2013]. The details of binary classification with probabilistic scores will be discussed in Chapter 3; the details of multi-class classification with probabilistic scores will be discussed in Chapter 5

2.5.2.2 Likert-scale labels In binary classification scenarios, Likert-scale labels are attached to traditional binary labels. Instead of either 0 or 1, the values of Likert-scale labels come as multiple Likert-scale levels. Such Likert-scale feedback also provides additional refined information regarding the class label compared with merely class label. For example, when diagnosing whether a patient is suffering from a disease, if the review is just a binary label indicating an infection, the patient may get lost whether to accept a therapy or to test further whether s/he is truly infected. However, if the review is a Likert-scale feedback, say 4 out of 6 indicating "probably infected", the patient may prefer to test further since the physician is not so confident of the infection; if the Likert-scale feedback is 6 out of 6 indicating "definitely infected", the patient may prefer to accept a therapy instantly. Therefore, we deeply expect higher performance from Likert-scale labels with the same number of annotated data samples. Likert-scale labels from the human annotators reflect the different degrees of certainty in binary labels. The notation of Likert-scale labels is based on Likert-scale, a universal multi-level rating scale for psychologic research, denoting the degrees of certainty in ordinal levels. For example, if a surgeon determines the probability of the emergence of some disease is 70%, the Likert-scale label of this event at a five-level Likert-scale scale from 0 to 4 should be 3. Another foundation for the intuition of Likert-scale labels is the consideration of cost. It is widely believed that a human annotator will usually assess a data sample comprehensively with splitting a data sample into multiple aspects for further assessments and weighing them together for a final evaluation. In other words, even when a human annotator is executing a mission of only a binary annotation, a comprehensive evaluation, which is very similar to a Likert-scale label, has already formed in mind. Thus, the cost of additional Likert-scale label annotation for traditional binary labels is negligible.

In this thesis, we propose new binary classification models with Likert-scale labels. We use

similar techniques with probabilistic scores: the Likert-scale labels naturally split the range of probabilistic scores into multiple consequent and non-overlapping bins, and naturally provide consistent orderings for learning an ordinal regression model. The details of binary classification with Likert-scale labels will be discussed in Chapter 4.

2.5.2.3 Ordered class set (OCS) Ordered class set (OCS) makes sense in multi-class classification scenario. Basically, an OCS defines an ordered subset of classes that represent choices that are likely (considered) for labeling the instance and their priority. An OCS may vary in size and includes classes that are considered to be viable class alternatives. Classes not in the OCS are considered to be unimportant or negligible. For example, in a four-class scenario, the OCS $\langle 3, 4 \rangle$ indicates that the annotator believes class 3 to be the most likely and class 4 to be the second most likely choice, while other two classes 1 and 2, are unimportant. Such ordered class sets provide more information indicating the weak connections between the instance and the alternative classes. For example, in an animal recognition task where each instance is a silhouette of an animal, if the output is just a class label indicating this animal is a cat, people may not get any other information of this cat. However, if the output indicates this animal is a cat and still of some probability a tiger, people get the information that, this cat may be larger or stronger than common cats so the annotator set tiger as an alternative choice. Another example is differential diagnosis, if the diagnosis is just a class label showing the patient is suffering from disease A, the patient will obtain no information apart from disease A, and only test further or take a therapy regarding disease A. However, if the diagnosis also comes with an alternative choice that s/he might be suffering from disease B instead of disease A, the patient may also prefer to test further regarding disease B. The problem of learning multi-class classification models from OCS is a new open problem. In this thesis, we propose a multi-class classifier that learns from OCS in addition to class labels. That is, each data instance is associated with an OCS of likely classes in an descending order regarding their relevance to the data instance.

In this thesis, we propose new multi-class classification models with OCS. We start by first defining and formalizing the problem of learning from OCS in multi-class settings upon AMSVM. After that, we present an algorithm for learning the multi-class classification model from such OCS feedback. The details of multi-class classification with OCS will be discussed in Chapter 6.

2.5.2.4 Permutation subset Permutation subset makes sense in multi-class classification scenario. Instead of a label vector, each data instance is associated with a permutation subset, a totally ordered subset over all the labels indicating the total orderings of the relevant labels of this instance according to their confidences. The labels not in the permutation subset are considered irrelevant to the instance. More formally, the permutation subset $S^{(i)}$ reflects the rankings of the relevant labels in terms of their importance to the instance among all the K labels. The permutation subset $S^{(i)}$ is formed by a non-empty subset of K labels indicating the descending ordering of the relevant labels. The labels not in the permutation subset are considered irrelevant to the instance by the annotator. For example, in a 4-label setting, a permutation subset $\langle 3, 2 \rangle$ indicates the 3rd label is the most relevant to the instance, the 2nd label is the second most relevant, and the other two labels are irrelevant. Such permutation subsets provide more information indicating the strongness of the connections between the instance and the labels. For example, when recognizing the genres of a song, if the output is just a label vector indicating this song is both rock and jazz, people may get confused how to credit its melody and pitches. However, if the output is a permutation subsets indicating this song is obviously rock while may also be jazz, people may tend to credit its melody and pitches as a rock music while only slightly credit its jazz element just as a spice. The problem of learning multi-label classification models from permutation subsets is a new open problem. In this thesis, we propose a two-stage multi-label ranking pipeline that learns from permutation subsets with two stages: a multi-label classifier finding the relevant labels and the dependencies among the labels, and an auxiliary multi-label ranker which ranks the relevant labels. The details of our two-stage multi-label ranking pipeline incorporating permutation subsets will be discussed in Chapter 7.

3.0 Active Learning of Binary Classification Models from Probabilistic Scores

3.1 Introduction

The work covered in this chapter was accepted and published in the 2017 conference of Florida Artificial Intelligence Research Society (FLAIRS) [Xue and Hauskrecht, 2017b]. In this chapter, our solution to reduce annotation effort on binary classification models seeks to advance a relatively new machine learning approach proposed to address the sample annotation problem: learning with probabilistic scores [Nguyen et al., 2011a, Nguyen et al., 2011b], in which each instance is associated with a probabilistic score reflecting the certainty or belief of human annotators in the specific class label, such as, a probability the patient suffers from a specific disease. A more ubiquitous example is the assessment of students, where such assessment in percentage can be treated as a probabilistic score. The benefit of probabilistic scores is that they let us distinguish data instances that are strong, weak or marginal representatives of a class, and when properly used in the classification training phase they can help us learn better classification models with a smaller number of labeled samples.

However, the caveat of learning from such probabilistic scores is that humans are unable to give consistent probabilistic assessments; a phenomenon well documented in psychology and decision making literature [Juslin et al., 1998, Griffin and Tversky, 1992]. In such a case, learning methods that are robust to "noisy" probabilistic scores are necessary. [Nguyen et al., 2011a, Nguyen et al., 2011b, Nguyen et al., 2013] address the problem by using probabilistic scores to first determine the relative order of examples in the training data and then build the final classification model by considering all pairwise orderings among them [Joachims, 2002, Herbrich et al., 1999]. They showed this approach is more robust to the noise in probabilistic scores than regression methods trying to directly fit probabilities. However, the limitations of their approach is that (1) the number of pairwise orderings one aims to satisfy is quadratic in the number of data points in the training data, and (2) all orderings are treated equally, that is, orderings induced by data points with smaller differences in probabilistic scores are treated equally to orderings with

larger differences.

The annotation effort can be further reduced via active learning [Lewis and Gale, 1994, Settles, 2010, Roy and McCallum, 2001] in which data instances are annotated sequentially one-by-one. Briefly, by smartly choosing the examples to be annotated next we expect to reduce the number of examples necessary to train a high quality classification model. In general, we seek to annotate the most informative data instance that helps to improve the quality of the classification model. While there are many different strategies for scoring and selecting unlabeled instances in the common binary classification models these either do not apply or are not optimized for probabilistic scores. We propose expected model change (EMC) strategy for binary classification models from probabilistic scores, which estimates the expected change on the prediction of the model for each unlabeled instance and possible probabilistic scores, then use it to select data instances that may help the model the best. To prevent the re-training of "add-one" models when adding an unlabeled instance and a possible probabilistic score into the labeled data, which is typically inefficient and required for traditional EMC strategy, we also train the add-one models incrementally from the current model rather than from scratch, which remarkably reduces the time consumption.

In this chapter, we first show how one can modify the all-pair problem formulation through binning where constraints within each bin are ignored and only constraints among data points in the different bins are enforced. This leads to a smaller number of pairwise constraints to satisfy and exclusion of constraints that are more likely corrupted by the noise. Second, we reformulate the problem of satisfying constraints among data points in different bins as an ordinal regression problem and solve it using ranking-SVM [Joachims, 2002, Herbrich et al., 1999] defined on these bins [Chu and Keerthi, 2005]. This reformulation further reduces the number of constraints one has to satisfy leading to a more efficient solutions where the number constraints to satisfy is linear in the number of samples.

3.2 Methodology

We start by first defining and formalizing our learning problem. After that we review an algorithm proposed by [Nguyen et al., 2011a, Nguyen et al., 2011b] for learning the binary classification model from data enriched with probabilistic scores, and gradually modify it to make it (1) more robust to noise and (2) more efficient to solve. Finally, we propose an expected model change active learning strategy and combine it with our binary classification model.

3.2.1 Problem description

Our objective is to learn a binary classifier $f : X \rightarrow Y$, where X is an input (feature) space and $Y = \{0, 1\}$ represents class labels one can assign to individual input instances. We want to learn the classifier, starting from an unlabeled dataset D_U that consists of input instances only. The labels to examples are assigned by a human annotator. In this chapter, we assume that in addition to binary $\{0, 1\}$ labels defining Y we also obtain probabilistic score: a probability p_i reflecting annotator's belief the example $\mathbf{x}_i$ belongs to class 1. Hence each labeled data entry d_i we can learn from consists of three components: $d_i = (\mathbf{x}_i, y_i, p_i)$, an input, a class label and an estimate of the probability of class 1. For example, if $\mathbf{x}$ is a patient and y denotes the presence or absence of a disease or some adverse condition that is based on physician's evaluation of the patient, the probability p_i captures the physician's belief the patient indeed suffers from the condition. The human-label assessment, especially the probabilistic score part, may not be perfect. This problem is well documented and was discussed in Section 2.5.2.1.

3.2.2 Method for learning with probabilistic scores

The approach in this chapter is motivated by the model proposed by [Nguyen et al., 2011a, Nguyen et al., 2011b] that is more robust to the noise in probabilistic scores. Briefly, instead of fitting the precise probabilities, it models the relation between probabilistic assessments in terms of pairwise order constraints of any two data entries in the labeled data, and uses them to drive the construction of a binary classifier.

To explain the approach in more depth, let us consider a function $f(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i$ allowing us

to discriminate between data entries of class 0 and class 1 after picking an appropriate threshold value. Using the probabilistic scores one way we can learn the function is by fitting the examples and probabilistic labels directly via regression. However, because regression is sensitive to the noise in probabilistic scores, [Nguyen et al., 2011a, Nguyen et al., 2011b] propose to learn this function from pairwise constraints induced by the probabilities. Briefly, if any two data entries $\mathbf{x}_j$ and $\mathbf{x}_k$ in the training data satisfy $p_j > p_k$, we expect the ordering function will preserve the order, that is $f(\mathbf{x}_j) > f(\mathbf{x}_k)$ or $f(\mathbf{x}_j) - f(\mathbf{x}_k) = \mathbf{w}^T(\mathbf{x}_j - \mathbf{x}_k) > 0$. The approach in [Nguyen et al., 2011a] aims to satisfy (pairwise) constraints for all pairs of examples in the training data. Since in practice some constraints may be violated, the authors' limit the number of pairwise constraint violations by using the pairwise-constraint loss function that is incorporated in the following optimization problem for finding the discriminative model [Nguyen et al., 2011a]:

$$\min_{\mathbf{w},w_0,\eta,\xi} \frac{1}{2}||\mathbf{w}||_2^2 + B\sum_{i=1}^{N}\eta_i + C\sum_{j=1}^{N-1}\sum_{k=j+1}^{N}\xi_{j,k}$$

$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 - \eta_i \qquad \forall i$$

$$\mathbf{w}^T(\mathbf{x}_j - \mathbf{x}_k) \geq 1 - \xi_{j,k} \qquad \forall j,k(p_j > p_k)$$

$$\eta_i, \xi_{j,k} \geq 0 \qquad \forall i,j,k$$

where $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, N-1$ and $k = j+1, j+2, \ldots, N$ index entries in the training data. w_0 defines the bias term and together with $\mathbf{w}$ it defines the binary decision boundary for the model. The first term in the objective function: $\frac{\mathbf{w}^T\mathbf{w}}{2}$ defines a regularization penalty, the second term (single sum) defines the hinge loss for all examples and their binary labels, and the third term (double sum) defines the pairwise-constraint loss function for pairs of probabilistic scores. η_i are slack variables defining the hinge loss, and $\xi_{j,k}$ slack variables reflecting individual constraint violation penalties for probabilistic score pairs $p_j > p_k$. Finally B and C are constants weighting the different loss and regularization terms in the objective function. The optimization will find the weights $\mathbf{w}$ and w_0 and the corresponding discriminant function that violates the minimum constraints.

3.2.3 Reducing the number of constraints via binning

The number of probabilistic score constrains in the above problem formulation is $O(N^2)$, more precisely $\frac{N(N-1)}{2}$. This negatively affects the efficiency of its solution. In this work we study binning to alleviate the problem.

The gist of the binning approach is that we divide the instances into several consequent, non-overlapping bins according to their probabilistic scores. The constraints for pairs of instances that fall into the same bin are then ignored; the constraints among instances in different bins are kept. One reason for applying this approach is that by binning we are more likely to remove constraints for instances with smaller probabilistic score differences, while preserving constrains for instances with larger probabilistic score differences. This is important since the noise in probabilistic scores (due to human variation in probabilistic score assessment) is more likely to flip the order of instances with small probabilistic score difference than the order of instances with larger probabilistic score difference. Hence the net effect of the binning is (1) the reduction in the number of constraints, as well as, (2) the selection of constraints that are more likely to be correct in terms of instance ordering. However, we would like to note that even with binning, the number of pairwise constraints in the formulation remains quadratic or $O(N^2)$. In the following, we develop a more efficient solution based on the ordinal regression that significantly improves the number of constraints one has to satisfy while learning the model.

The idea of binning is to satisfy constraints only among entries placed in the different bins. Optimally we would like to have data entries that are in the same bin according to its probability label fall into the same bin also after the projection. We can use this intuition to reformulate the optimization problem as an ordinal regression problem [Chu and Keerthi, 2005]. Briefly we want to find the function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ that puts the data points into bins according to their probabilistic scores. We can achieve this by having every example $\mathbf{x}$ project on the correct side of each bin boundary. For example, if the example $\mathbf{x}$ is located in ith bin, then after the projection, $f(\mathbf{x})$ should be smaller than the lower margin (boundary) of bin j in the projected space, whenever $i < j$. In general, assuming m bins labeled from 1 to m, bin boundaries $b_1, b_2, \ldots b_{m-1}$ separating them in the projected space, and bin function $bin(p_i)$ that maps the probability to the bin number (lowest probability maps to lowest number), then, after the projection, the example x_i with probabilistic

score p_i should project to value smaller than b_j whenever $bin(p_i) \leq j$, otherwise its value should be larger than b_j. Overall, for N data entries and m boundaries there are $(m-1)N$ constraints, one for each data entry/boundary pair. To guarantee the robustness of our model against noise in probabilistic scores, we allow violations of constraints by penalizing the loss function of sample/boundary pairs. By adding the constraints for binary class labels, we can formulate the following optimization problem:

$$\min_{\mathbf{w},w_0,\mathbf{b},\boldsymbol{\eta},\Xi} \frac{1}{2}||\mathbf{w}||_2^2 + B\sum_{i=1}^{N}\eta_i + C\sum_{j=1}^{m-1}\sum_{i=1}^{N}\xi_{j,i}$$

$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 - \eta_i \qquad \forall i$$

$$\mathbf{w}^T\mathbf{x}_i - b_j \leq \xi_{j,i} - 1 \qquad \forall i, j(bin(p_i) \leq j)$$

$$\mathbf{w}^T\mathbf{x}_i - b_j \geq 1 - \xi_{j,i} \qquad \forall i, j(bin(p_i) > j)$$

$$\eta_i, \xi_{j,i} \geq 0 \qquad \forall i, j$$

where $j = 1, 2, ..., m-1$ indexes bin boundaries in $\mathbf{b}$, and $i = 1, 2, ..., N$ indexes data entries. The first term in the objective function is the regularization term, the second term (single sum) defines the hinge loss with respect to binary labels, and the third term (double sum) defines the bin-constraint loss function. η_i and $\xi_{j,i}$ are slack variables permitting violations of binary class and probabilistic score bins respectively. B and C are constants weighting the objective function terms. Again, this optimization yields a discriminant function $f(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i + w_0$ that tries to minimize the number of violated constraints, but the number of constrains is reduced to $O(mN)$ as compared to $O(N^2)$ for the pairwise-ordering methods (with or without the binning).

3.2.4 Choosing the best bin number

One question that remains open is how to define bins and how to choose their number. To answer this question, let us first revisit our ordinal-regression-based method and pairwise-ordering-based classifier.

In our pairwise-ordering-based method, for a probabilistic score, we enforce the pairwise ordering between it and each probabilistic score. In our ordinal-regression-based method, for a

probabilistic score, we enforce the pairwise ordering between it and each bin boundary. Therefore, ordinal-regression-based method can be treated as an approximation of pairwise-ordering-based method: the pairwise-ordering-based method still considers each probabilistic score in the same bin, while the ordinal-regression-based method considers all the probabilistic scores in the same bin as an entity. In other words, pairwise-ordering-based method still considers the probabilistic-score distribution of each bin, while the ordinal-regression-based method approximates the probabilistic-score distribution of each bin as a uniform distribution. That is, the ordinal-regression-based method approximate the probabilistic-score distribution in the same way as histogram. The relations among pairwise orderings, ordinal regression, actual and histogrammed probabilistic-score distribution are illustrated in Figure 4.

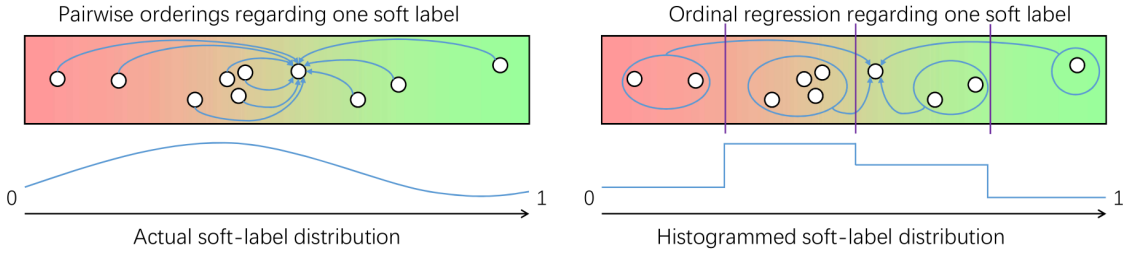

Figure 4: Relations among pairwise orderings, ordinal regression, actual and histogrammed distribution on probabilistic scores (soft labels).

One possible solution to define the bins is to use an equal-distance binning that splits the range of values (in our case probabilistic-score values) equally. Another possibility choose bins of equal size. Since the ordinal-regression-based method approximates the probabilistic-score distribution of each bin as a uniform distribution, equal-distance binning may not be a good choice: if there are too few examples in one bin, the probabilistic-score distribution of this bin may not be well estimated. Therefore, in this work, we use equal-size binning, that is, the bin boundaries are built such that each bin covers approximately the same number of examples.

Another challenge is to choose the number of bins. The caveat here is that the number of bins may affect the quality of the result. If we use $N - 1$ bins where each bin only contains one data sample, the optimization problem is similar to our pairwise-ordering-based method with $O(N^2)$ constraints. On the other hand, if we only use two bins, the bin/sample pairwise ordering

is equivalent to binary classification. The optimal bin choice is somewhere in between these two extremes. One approach to select the number of bins is to use the internal cross-validation approach. Another is to use a heuristic function. Since the ordinal-regression-based method approximate the probabilistic-score distribution in the same way as histogram, it is desired that the approximate distribution after binning is close to the actual probabilistic-score distribution. This can be achieved by Freedman-Diaconis rule [Freedman and Diaconis, 1981] for histogram, which minimizes the mean squared difference between the histogram distribution and true data distribution. Briefly, [Freedman and Diaconis, 1981] determines that the number of bins for N examples should follow $floor(\sqrt[3]{N})$ trend. In this subsection, we analyze this heuristic function and compare it to the internal cross-validation approach.

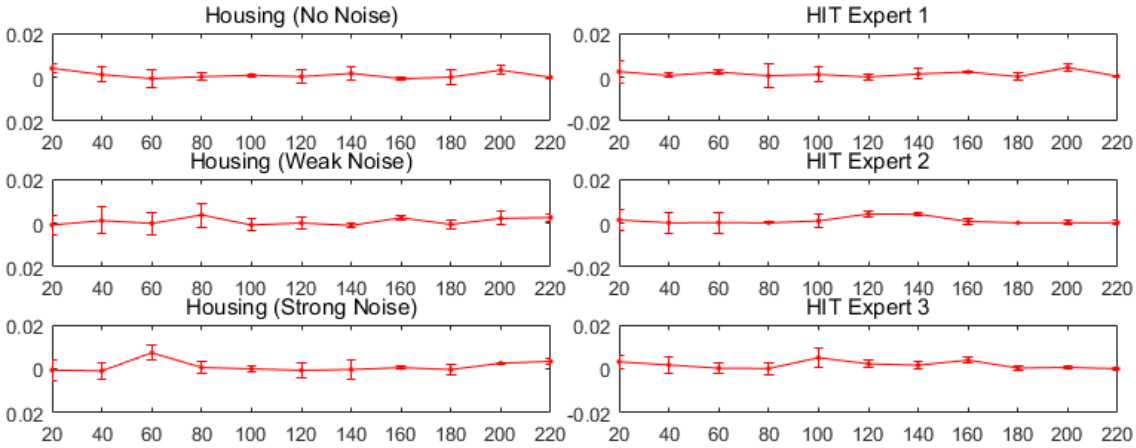

Figure 5: Average AUROC difference for two versions of the ordinal-regression-based method on six datasets.

To show how close these two approaches are, Figure 5 plots average differences in AUROC scores for the cross-validation and heuristic approaches on the housing data (with three levels of noise) and three HIT datasets. Clearly the differences in performance across all these experiments are very small, suggesting the our heuristic function based on [Freedman and Diaconis, 1981] a good choice for determining the number of bins. In the remaining parts of this paper, we always apply this heuristic function to find the optimal bin number for our ordinal-regression-based method.

3.2.5 Active learning

The next challenge is to embed the above learning algorithm in a practical active learning framework. The heart of any active learning method is a strategy that is used to select examples to be queried next. In this work, we propose and experiment with a strategy called expected model change (EMC) [Tong and Koller, 2000, Settles et al., 2008b] that evaluates and measures the potential of an unlabeled data instance to change the model by estimating its impact on instance predictions. Our expected model change (EMC) also uses a Bayesian posterior to calculate the expectation.

We can let the unlabeled instance to be added into the current model as $\langle \mathbf{x}^+, y^+, p^+ \rangle$ triplet, where p^+ is one possible probabilistic score the example can be assigned to and y^+ is the corresponding class label. However, in this subsection, since the classification model are trained on m bins rather than exact values, the N different probabilistic scores from the training data actually split the range of the probabilistic scores $[0, 1]$ into m ordinal categories; the discriminant hyperplane of binary classes also splits the probabilistic score range into two ordinal categories. Therefore, the probabilistic score range can be split into $m + 1$ ordinal categories, where any probabilistic score p^+ in the same category have the same class label y^+ and the same pairwise ordering relationship with the existing $m-1$ bin boundaries in the training data, eventually leading to the same add-one model. In this section, instead of focusing on the exact value of p^+ and y^+, we only need to focus on the corresponding ordinal category c^+ of p^+ and y^+.

To select the unlabeled instance with the highest expected model change, we need to propose efficient calculation of two quantities: (1) the model change when given an ordinal category; (2) the expectation of model change over the joint distribution of the class label and probabilistic score. To prevent the re-training of "add-one" models when adding an unlabeled instance and an ordinal category into the labeled data, which is typically inefficient and required for traditional EMC strategy, we also train the add-one models incrementally from the current model rather than from scratch, which remarkably reduces the time consumption.

3.2.5.1 Expected model change Briefly, the expected model change (EMC) [Tong and Koller, 2000, Settles et al., 2008b] of an unlabeled sample $\mathbf{x}$ can be measured as

follows: Suppose that, for the labeled data L, we have already trained a model f_L. For an unlabeled instance $\mathbf{x}^+$, there are $m+1$ possible ordinal categories. For each possible ordinal category c^+, if we add $\langle \mathbf{x}^+, c^+ \rangle$ into L, we will obtain an add-one model $f_{L \cup \langle \mathbf{x}^+, c^+ \rangle}$. The model change of $f_{L \cup \langle \mathbf{x}^+, c^+ \rangle}$ compared with f_L is denoted as $\delta(\mathbf{x}^+, c^+)$. Since there are $m+1$ possible ordinal categories, we will have $m+1$ performance changes $\delta(\mathbf{x}^+, c^+)$ where $c^+ = 1, 2, \ldots, m+1$, each corresponding to one add-one model. The expected model change $\Delta(\mathbf{x}^+)$ is then calculated as:

$$\Delta(\mathbf{x}^+) = \sum_{c^+=1}^{m+1} P(c^+|\mathbf{x}^+)\delta(\mathbf{x}^+, c^+)$$

3.2.5.2 Model change To measure the model change $\delta(L, \langle \mathbf{x}^+, c^+ \rangle)$ for an unlabeled example $\mathbf{x}^+$ and one possible ordinal category c^+, we measure the model change as the discrepancy of the categorical predictions $||cat[g(\mathbf{x}_j)] - cat[g^+(\mathbf{x}_j)]||$, where $cat(\cdot)$ returns the ordinal category ranging in $\{1, 2, \ldots, m+1\}$ of the model output; $g(\cdot)$ and $g^+(\cdot)$ are the models before and after $\langle \mathbf{x}^+, c^+ \rangle$ are added, respectively. By considering every unlabeled example, the model change $\delta(L, \langle \mathbf{x}^+, c^+ \rangle)$ can be calculated by summing its impact over the unlabeled data as:

$$\delta(L, \langle \mathbf{x}^+, c^+ \rangle) = \sum_{j \in U} ||cat[g(\mathbf{x}_j)] - cat[g^+(\mathbf{x}_j)]||$$

where j indexes all examples in the unlabeled data U.

3.2.5.3 Distribution of ordinal categories After calculating performance changes $\delta(L, \langle \mathbf{x}^+, c^+ \rangle)$ for all possible ordinal categories c^+, we also adopt a Bayesian method to estimate the distribution of $m+1$ ordinal categories. More formally, let $\mathbf{x}^+$ be an unlabeled instance we are considering to query and $k = cat[g(\mathbf{x})^+]$ be the predicted ordinal category the instance falls into based on g. Our objective is to estimate the probability distribution $(P_1^k, P_2^k, \ldots, P_{m+1}^k)$ for category k, which represents the probability of an example predicted in category k to be actually labeled to one of the $m+1$ categories. One way to estimate this probability would be to use the maximum likelihood approach and calculate the probabilities from

counts of labeled categories $q_1^k, q_2^k, \ldots, q_{m+1}^k$ in L that are predicted to fall into category k, and by assuming they follow a multinomiall distribution with parameters $(P_1^k, P_2^k, \ldots, P_{m+1}^k)$. We estimate the posterior distribution of $(P_1^k, P_2^k, \ldots, P_{m+1}^k)$ to prevent the bias. we use a Dirichlet prior which is the conjugate choice for the multinomial distribution. Since we do not have any prior information about the distribution of labeled categories in the predicted category, we choose $Dirichlet(1, 1, \ldots, 1)$ where all labeled categories are assigned the same prior probability. Given the conjugate prior, the posterior of $(P_1^k, P_2^k, \ldots, P_{N+2}^k)$ follows a Dirichlet distribution:

$$(P_1^k, P_2^k, \ldots, P_{m+1}^k) \sim Dirichlet(1 + q_1^k, 1 + q_2^k, \ldots, 1 + q_{m+1}^k)$$

Given the posterior distribution, we can approximate the probability $P(c^+|\mathbf{x}^+)$, that is, the probability that $\mathbf{x}^+$ is assigned label c^+, by the expected value of $E(P_{c^+}^k)$ from the posterior distribution:

$$P(c^+|\mathbf{x}^+) = E(P_{c^+}^k) = \frac{1 + q_{c^+}^k}{m + 1 + \sum_{l=1}^{m+1} q_l^k}$$

3.2.5.4 Incremental training of add-one models Another critical question is the running time complexity to obtain an add-one model g^+ after adding an unlabeled sample $\mathbf{x}^+$s and possible ordinal category c^+ into labeled data. In this section, we adopt the sequential minimal optimization algorithm by [Platt, 1999] for our SVM-based classification models. If we train the add-one model from scratch, the time complexity is $O(K^2)$ where $K = mN$ is the number of constraints, which is linearly proportional to the bin number and labeled instance number. Since, in order to select the unlabeled example to be labeled next, we need to obtain an add-one model for each unlabeled example and each possible ordinal category label, the total time complexity is $O(K^2 m|U|) = O(m^3 N^2 |U|)$ ($|U|$ is the size of the unlabeled data), which is extravagant and does not scale well as the size of N grows. To solve this problem, instead of starting from scratch, we always start from the current model g to train the add-one model g^+, which remarkably reduces the time complexity of training one add-on model into $O(K) = O(mN)$. Therefore, the total time complexity of training all the add-one models for the current model g is reduced to $O(Km|U|) = O(m^2 N|U|)$.

3.3 Experiments and results

We test our approach on both synthetic and real-world data. The first set of experiments uses data from several UCI regression data sets which we transform to probabilistic score problems. We use these data to show the performance of the methods when probabilistic scores are corrupted with the different level of noise. The second experiment works with real-world clinical data with true (human assessed) probabilistic labels. Finally, we test our active learning strategy on synthetic data.

3.3.1 Experiments of probabilistic scores on synthetic UCI data

In this part we adapted one UCI regression data set (Housing) and three UCI ordinal classification data sets (Cancer, Wine Red, Wine White) as follows. For the UCI housing regression data set we normalized the outputs ranging in R and reinterpreted them as probabilistic scores. We also defined a binary class threshold over the probabilistic scores to distinguish class 0 from class 1. For example, the outputs in Housing data set represents the attractiveness of houses to the consumers. In this case, we define two classes: houses with high attractiveness (class 1) and houses with low attractiveness (class 0). We use 30% of data entries with top score to define class 1, the rest are assigned to class 0. The UCI ordinal classification data sets come with multiple classes and full-order relations among classes. We generate probabilistic labels by evenly normalizing the class labels according to the total number of classes. The binary thresholds can be set according to the meaning of ordinal classes. For example, Breast Cancer data set contains six ordinal classes $\{1, 2, 3, 4, 5, 6\}$, where $\{1, 2\}$ are healthy and $\{3, 4, 5, 6\}$ represent the different stages of malignancy. We use this information to re-map the class labels into $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ with a threshold of 0.3 for the binary label.

SoftSvmOrd: the SVM-based ordinal regression method derived from the orderings between probabilistic scores and bin boundaries. The probabilistic scores are split into m bins from our optimal binning scheme. The next unlabeled instances to be labeled are selected randomly;

SoftSvmOrdAct: the SVM-based ordinal regression method derived from the orderings between soft labels and bin boundaries. The soft labels are split into m bins from our optimal

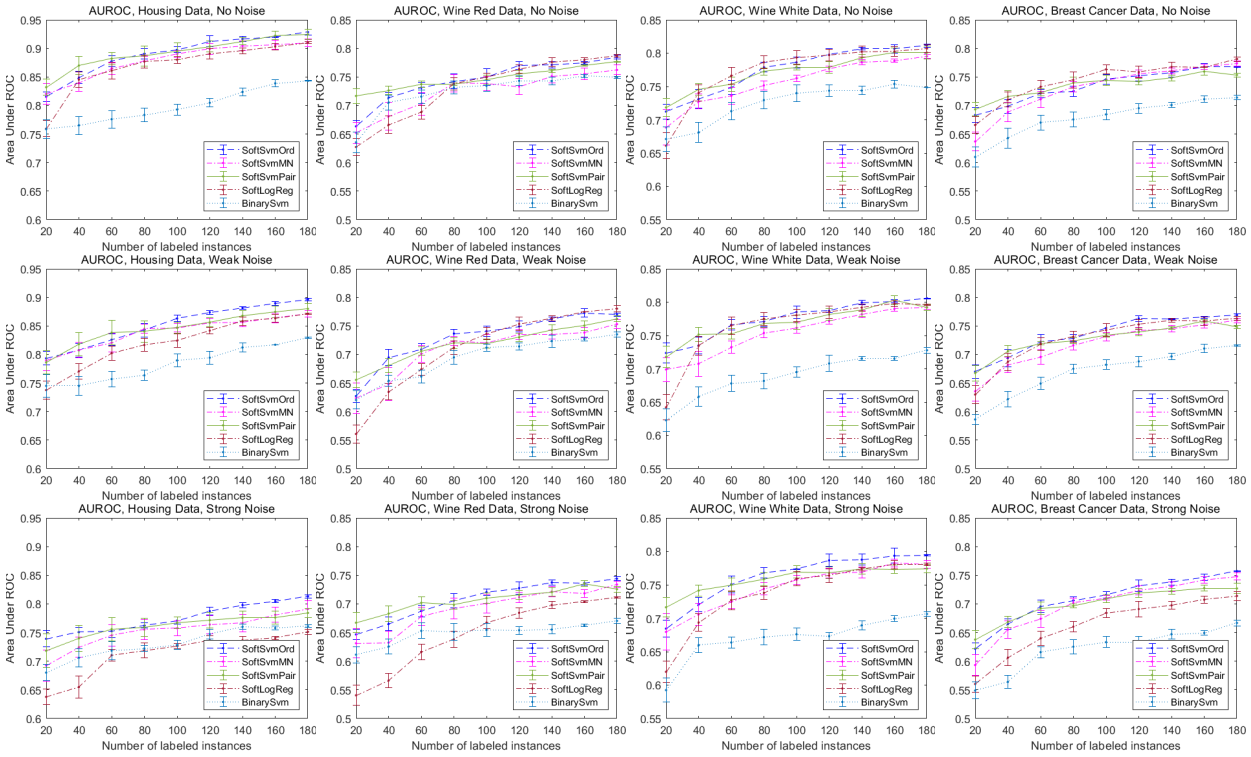

Figure 6: Performance with random sampling on four synthetic datasets regarding different labeled instance numbers with no (top), weak (middle) and strong (bottom) noise.

binning scheme. The next unlabeled instances to be labeled are selected from our expected model change strategy for binary classification with probabilistic scores;

SoftSvmPair: the SVM-based ranking method derived from the pairwise orderings of probabilistic scores. The next unlabeled instances to be labeled are selected randomly;

SoftSvmPairAct: the SVM-based ranking method derived from the pairwise orderings of probabilistic scores. The next unlabeled instances to be labeled are selected from our expected model change strategy for binary classification with probabilistic scores;

SoftSvmMN: the SVM-based ranking method derived from the pairwise orderings of probabilistic scores, where only mN random pairwise orderings are enforced. Therefore the number of constraints is the same as **SoftSvmOrd**. The next unlabeled instances to be labeled are selected randomly;

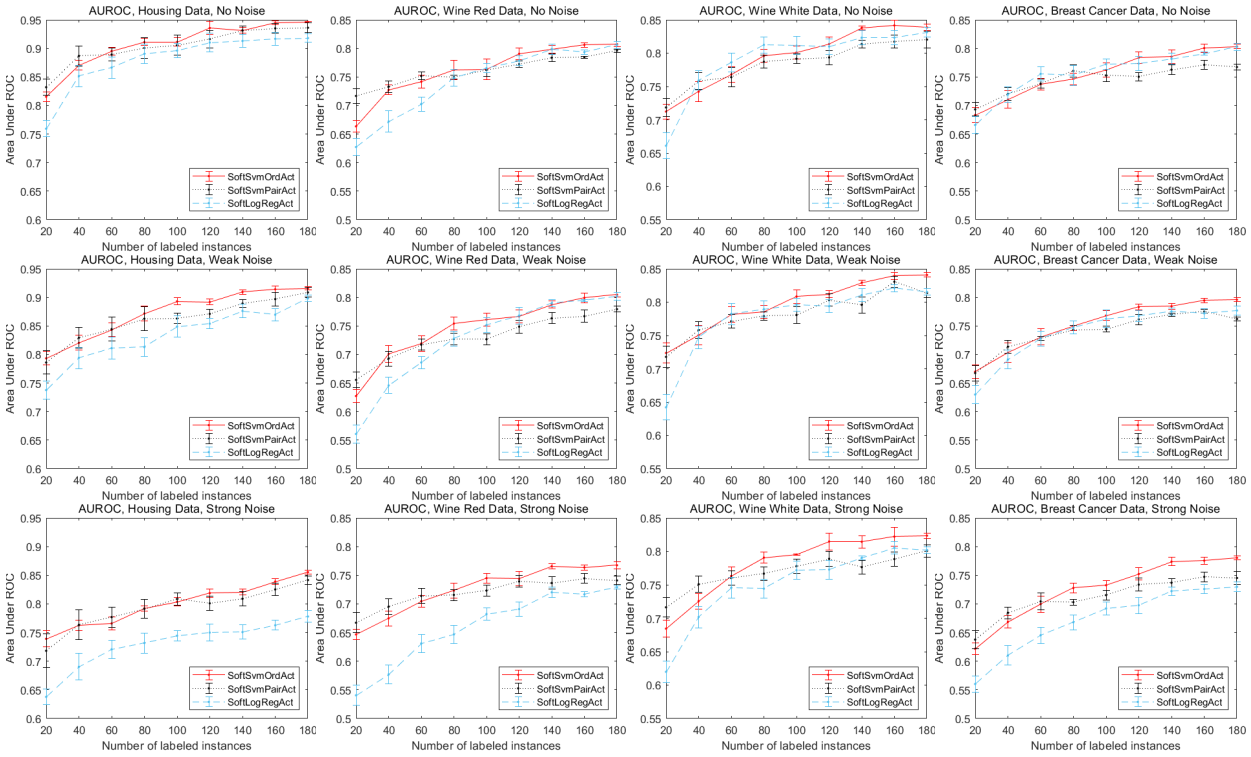

Figure 7: Performance with active learning on four synthetic datasets regarding different labeled instance numbers with no (top), weak (middle) and strong (bottom) noise.

SoftLogReg: the regression-based model derived from the exact values of probabilistic scores. The next unlabeled instances to be labeled are selected randomly;

SoftLogRegAct: the regression-based model derived from the exact values of probabilistic scores. The next unlabeled instances to be labeled are selected from our expected model change strategy for binary classification with probabilistic scores;

BinarySvm: the SVM model trained on binary labels only. The next unlabeled instances to be labeled are selected randomly;

We evaluated the performance of the different methods by calculating the Area under the ROC (AUC) the learned classification model would achieve on the test data. Hence, each data set prior to the learning was split into the training and test set (using $\frac{2}{3}$ and $\frac{1}{3}$ of all data entries respectively). The learning considered training data only, the AUC was always calculated on the test set. The test

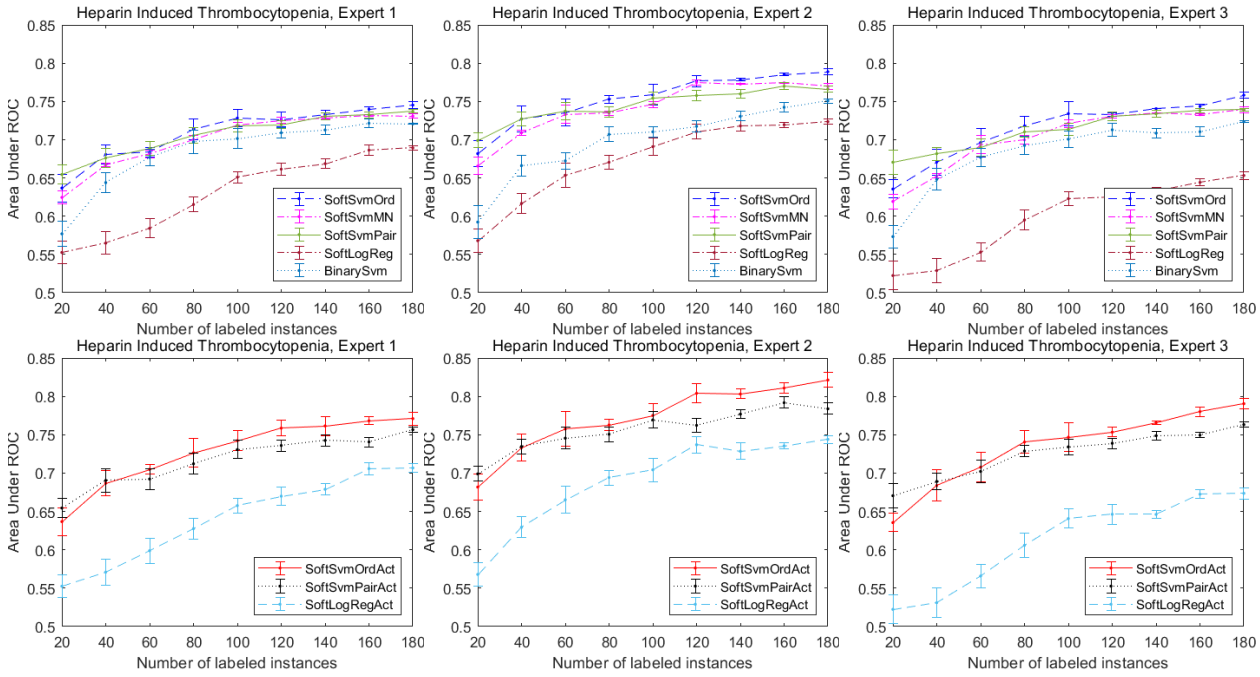

Figure 8: Performance on real-world HIT dataset annotated by three experts regarding different labeled instance numbers.

set performance reflects how well the model generalizes to future data. To avoid potential train/test split biases, we repeated the training process (splitting) and learning steps 24 times. We report the average AUC obtained on these test sets. To test the benefits of our active learning strategy and the impact of probabilistic scores on the number of data entries, we trace the performance of all models for the different sizes N of labeled data. Figure 6 shows the performance (AUC) of the models on all four UCI data sets for increasing sizes of N and the different levels of noise in probabilistic scores.

3.3.1.1 Benefit of probabilistic scores and active learning Figure 6 (top) and 7 (top) show the performance of methods when simulated probabilistic scores are not corrupted by additional noise. Among the methods without active learning, all the four probabilistic-score-based methods, *SoftSvmOrd*, *SoftSvmPair*, *SoftSvmMN* and *SoftLogReg*, outperforms *BinarySvm* which only utilizes binary labels. This demonstrates the sample-size benefit of probabilistic scores

for learning classification models when the probabilistic scores provided are accurate without noise. Also, *SoftSvmOrdAct* outperforms *SoftSvmOrd*; *SoftSvmPairAct* outperforms *SoftSvmPair*; *SoftLogRegAct* outperforms *SoftLogReg*. These three comparisons validate the effectiveness of our expected model change active learning strategy for binary classification with probabilistic scores. Overall, *SoftSvmOrdAct*, *SoftSvmPairAct* and *SoftLogRegAct* are the top three of all the methods, showing the benefit of combining probabilistic scores and active learning. These three methods combining probabilistic scores and active learning perform comparably well. Similarly, out of all probabilistic-score-based methods without active learning, there does not seem to be a clear winner and all methods perform comparably well. Please notice that *SoftLogRegAct* and *SoftLogReg* methods which fit the exact probabilities to the model via regression is comparable to other methods.

3.3.1.2 Effect of noise on probabilistic scores Figure 6 (top) and 7 (top) results assumed the probabilistic scores directly reflect the probabilistic information. However, in practice, probabilistic information (when collected from humans) may be imprecise and subject to noise. This in turn may affect the quality of our models. Our synthetic noise experiments aim to show the robustness of the methods to noise in probabilistic scores. In order to generate noise in probabilistic scores, each probabilistic score p derived from the UCI data, was modified into p' by injecting a Gaussian noise of different strength:

Weak noise: $p'_i = p_i \times (1 + 0.1 \times \alpha_i)$

Strong noise: $p'_i = p_i \times (1 + 0.3 \times \alpha_i)$

where α_i is a random variable that follows standard normal distribution $\mathcal{N}(0, 1)$. Briefly, the noise injection levels above indicate the average proportion of noise to signal at weak (10%) and strong (30%) levels respectively. Also, we truncated the illegal probabilistic scores (e.g. probabilistic score that are less than 0 or greater than 1) to the interval of $[0, 1]$. The results of the different methods for the weak and strong noise are summarized in the middle and bottom rows of Figure 6 and 7 respectively.

When noise is added into the probabilistic labels, in Figure 6 (middle, bottom) and 7 (middle, bottom), the performance of a model may drop. Two methods, *SoftLogReg* and *SoftLogRegAct*, that directly fit the exact probabilities are particularly sensitive to the noise and their performance

drops significantly for both noise levels and across all datasets. Other models that use pairwise orderings or instance/bin orderings derived from probabilistic scores are more robust and do not suffer from such a performance drop. Our new methods, *SoftSvmOrd* and *SoftSvmOrdAct*, are the most consistent and tends to outperform other SVM-based probabilistic-score models in both noise injection levels. Also, *SoftSvmOrdAct* outperforms *SoftSvmOrd*; *SoftSvmPairAct* outperforms *SoftSvmPair*; *SoftLogRegAct* outperforms *SoftLogReg*. These three comparisons validate the effectiveness of our complementary expected model change active learning strategy when noise is injected to probabilistic scores. Overall, our new model *SoftSvmOrdAct* combined with our expected model change strategy is the best on all datasets, showing the benefit of combining probabilistic scores and active learning when noise is injected. These experiments demonstrate the robustness of our methods on the learning tasks with probabilistic scores.

3.3.2 Experiments and results on time complexity

One of the reasons for introducing the new binning method was to improve the pairwise constraint solution (*SoftSvmPair* method) proposed by Nguyen [Nguyen et al., 2011a]. Figure 9 shows the time consumption of three probabilistic score methods used earlier (*SoftSvmPair*, *SoftSvmOrd* and *SoftSvmMN*) on UCI data sets for increasing sizes of N weak noise in probabilistic scores.

We evaluated the time consumption of the different learning methods by the total minutes elapsed on the training data. For *SoftSvmOrd* and *SoftSvmMN* we always keep the same number of probabilistic score constraints: KN. As expected, *SoftSvmOrd* and *SoftSvmMN* running times are very close across all experiments. In contrast to these, the performance of *SoftSvmPair* that uses all $\frac{N(N-1)}{2}$ pairwise constraints deteriorates very quickly as N increases, and at $N = 180$ the running time increases about four fold when compared to our *SoftSvmOrd* approach. This confirms the running-time benefit of *SoftSvmMN* and *SoftSvmOrd* with the reduced number of probabilistic score constraints. Please notice that the results in Figure 6 and in Figure 9 combined demonstrate the benefit of our new method *SoftSvmOrd*. It tends to outperform the baseline *SoftSvmPair* in terms of the solution quality across many sizes N and this with a remarkably lower running time. It also outperforms *SoftSvmMN* in terms of the solution quality at comparable running times. Overall,

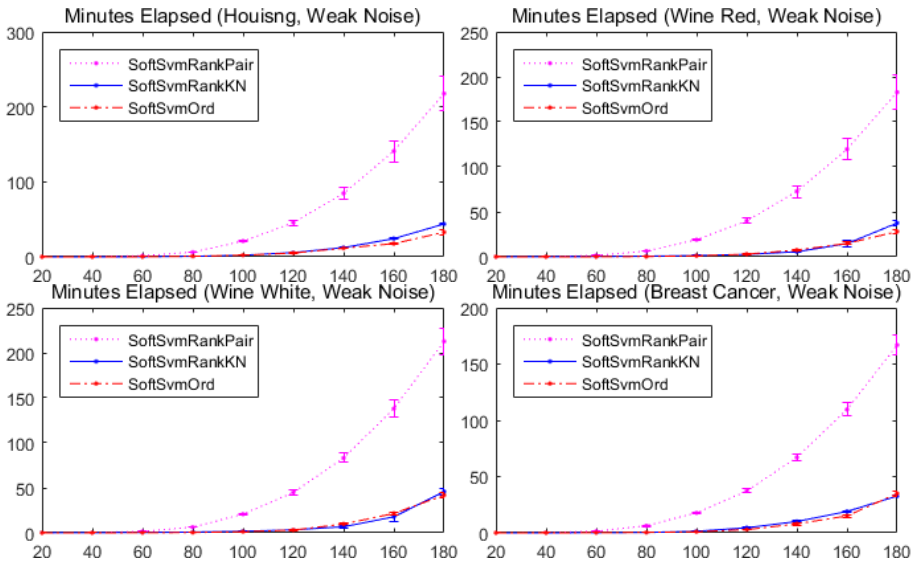

Figure 9: Time consumption (minutes) regarding different labeled instance numbers on four synthetic datasets with weak noise.

our learning methods for binary classification from probabilistic scores not only improves the predictive performance, but also reduce the time complexity.

3.3.3 Experiments of probabilistic scores on clinical data

While the experiments on synthetic data sets support the benefits of our probabilistic-score-based approach, it is unclear whether these results also extend and generalize to "true" probabilistic scores assessed by humans. In this set of experiments we test the performance of the methods on the real-world clinical data that were independently reviewed and assessed in terms of probabilistic scores by three different experts. The target label concerned Heparin induced thrombocytopenia (HIT), an adverse clinical condition that affects patient who are treated with heparin for prolonged periods of time. The clinical data consists of 50 patient state features important for detection of HIT derived from the PCP database [Hauskrecht et al., 2010, Valko and Hauskrecht, 2010, Hauskrecht et al., 2013]. The datasets consists of 579, 571, and 573 labeled patient state instances for Expert 1, 2 and 3,

respectively (see [Valizadegan et al., 2013]). The labels include both binary and probabilistic scores [Nguyen et al., 2011a, Nguyen et al., 2013].

Figure 8 shows the AUROC performance of the same methods and models as used in the previous section on three expert-annotated HIT datasets. On all three datasets the performance of our *SoftSvmOrdAct* method is the best and it outperforms all other methods. This experiment confirms good performance of our method and the benefit of combining probabilistic scores and active learning for more efficient training of binary classification models.

3.4 Summary

To obtain labels for classification purposes, we often rely on human annotators. However, the human annotation process may be costly. In such a case, different methods of reducing the labeling cost need to be applied. In this chapter we have developed and tested a new robust method that uses probabilistic scores that is able to enrich the feedback one receives from human and hence improve the number of examples one has to label to get a good classification model. Our results on synthetic and real-world clinical data show that our method (1) can benefit greatly from additional probabilistic scores (2) is robust to the different levels of noise in probabilistic scores.

4.0 Active Learning of Binary Classification Models from Likert-scale Feedback

4.1 Introduction

The work covered in this chapter was accepted and published in the 2017 conference of SIAM Data Mining (SDM) [Xue and Hauskrecht, 2017a]. In Chapter 3 we proposed and tested a new method based on binning of probabilistic scores to ordinal categories and enforced the constraints on these categories. In this Chapter we directly seek a feedback in terms of ordinal Likert-scale categories instead of probabilities.

Briefly, Likert-scale categories define a set of ordinal categories humans can use to provide information about the strength of agreement (or belief) in the respective class labels. For example, when obtaining a feedback from a physician on whether the patient suffers from a particular disease or not, the binary true/false feedback can be refined by obtaining physician's belief in the presence of the disease on a 5-point Likert scale by asking if he/she agrees, weakly agrees, is neutral, weakly disagrees, or disagrees with the disease. Another more ubiquitous example is the user reviews in online stores, where each review is associated with five-star assessment. In terms of Chapter 3 solutions we can see Likert-scale categories to be equal to be qualitative bins and the annotator is asked to assign examples to these bins directly. We also develop a new variant of expected model change (EMC) active learning strategy that attempts to optimize the example selection by considering the Likert-scale feedback. To prevent the re-training of "add-one" models when adding an unlabeled instance and a possible Likert-scale label into the labeled data, which is typically inefficient and required for traditional EMC strategy, we also train the add-one models incrementally from the current model rather than from scratch, which remarkably reduces the time consumption.

We test our new framework on multiple classification problems based on UCI and real-world clinical decision problem data. We demonstrate the ability of our solutions to reduce the data labeling cost both individually and in combination.

4.2 Methodology

In this part, we develop an active learning framework that builds a classification model by actively querying an annotator who provides feedback to the framework for assessing the instances using Likert-scale categories. We start by first defining and formalizing the problem of learning from ordinal Likert-scale category labels. After that, we present an algorithm for learning the classification model from such feedback. Second, we show how this algorithm can be included in the active learning framework that aims to improve the model by wisely selecting the examples to be assessed next. The criterion used to choose from among unlabeled candidate instances is based on the highest expected classifier prediction change. We also briefly describe solutions to modeling the distribution which is used to calculate the expected change. To prevent the re-training of "add-one" models when adding an unlabeled instance and a Likert-scale label into the labeled data, which is typically inefficient and required for traditional EMC strategy, we also train the add-one models incrementally from the current model rather than from scratch, which remarkably reduces the time consumption.

4.2.1 Problem settings

Our objective is to learn from data a binary classifier: $C : X \rightarrow Y$, where X is a feature space and $Y \in \{0, 1\}$ is one of the two class labels. At the very beginning, all the examples in dataset D are unlabeled. However, we can sequentially query a human annotator to provide information for individual examples and use this feedback to build a classification model. We assume that in addition to traditional binary labels $Y = \{0, 1\}$, each data example is also assessed in terms of ordinal Likert-scale categories characterizing the degree of agreement of the annotator in its assignment to one of the classes. Therefore, a labeled data sample d_i is a vector consisting of three parts $(\mathbf{x}_i, y_i, u_i)$, that is, a vector of features, a traditional binary label and a Likert-scale label indicating the level of agreement that the data example falls into one of the two classes. Both y_i and u_i are based on human annotator feedback. For example, if a human expert is asked to assess a patient whether he or she suffers from a particular disease, $\mathbf{x}$ represent the labs, symptoms and observations describing the patient state, y is expert's disease/no-disease decision, and u represents

the degree of which the expert believes in (and agree) with the disease diagnosis.

4.2.2 Learning a classifier from Likert-scale labels

Let us focus first on the task of learning a classification model from the data represented by the triplets $(\mathbf{x}_i, y_i, u_i)$, that is, we assume the data with this information are available and can be used. One way to learn the classification model would be to adapt and build upon approach proposed by [Nguyen et al., 2011a, Nguyen et al., 2011b] for probabilistic scores with noise. Briefly, their approach seeks to find a ranking function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}_i$ that aims to satisfy all pairwise constraints among data points ordered according to their 'noisy' probability estimate reflecting the confidence of the annotator in the binary class label. They formulate and solve the problem using an SVM-like optimization task that seeks to satisfy as many constraints as possible. The key trick in their approach is that the same ranking function can also be used to define a discriminative projection that lets us discriminate between class 0 and class 1 data instances. We can quickly adapt their approach and apply it to Likert-scale assessments by creating pairwise ordering constraints only among data points that fall into the different Likert-scale categories. Briefly if two data entries $\mathbf{x}_i$ and $\mathbf{x}_j$ in the dataset are assigned ordinal category labels such that $u_i > u_j$, we expect that the same order will be preserved also by the the ranking function: $f(\mathbf{x}_i) > f(\mathbf{x}_j)$. Similarly to [Nguyen et al., 2011a, Nguyen et al., 2011b], a classifier, and its discriminative projection can be then defined using the same ranking function.

Unfortunately, the above solution suffers from a drawback: the number of pairwise constraints one wants to satisfy grows quadratically with the number of data instances which negatively affects the time-complexity and scalability of the solution. To alleviate the scalability problem, we try to abridge the number of constraints imposed on the ordinal Likert-scale labels. In this chapter, we propose an improvement based on 'binning' of values of the ranking function f. The idea of the solution is that after the projection (via ranking function), all examples with the same ordinal category label should, in the ideal case, fall into the same value region or bin.

Let us assume that for each ordinal label u we have a bin defined by a lower bound value b_{u-1} and an upper bound value b_u. Our objective is to find a projection f from the feature space to the space of real numbers, for which instances that are in the same ordinal category fall after

the projection into the same bin. More formally, for any data instance $\mathbf{x}_i$ and its ordinal label u_i, we expect to obtain a function $f(\cdot)$ so that $bin(f(\mathbf{x}_i)) = u_i$, where $bin(\cdot)$ is a function where the argument is a prediction value and the return value is the bin where this prediction value belongs. Then each data example with ordinal label u should be projected such that its value is greater than all bin bounds b_j such that $j < u$ and less than all b_k such that $k \geq u$. Since the ordinal label and the projection of its feature vector to the bin are always expected to match, the projection should have the same greater-or-less relationship with all bin boundaries for other ordinal categories. Formally, for a data instance $\mathbf{x}$ and Likert-scale label u, we expect to learn a function so that $f(\cdot)$ so that $b_j < f(\mathbf{x})$ for any $j < u$ and $b_k > f(\mathbf{x})$ for any $k \geq u$.

However, in reality, we cannot expect that all the constraints will always be satisfied with a linear projection function. Hence, we permit violations of constraints but penalize them via bin-sample loss function. By adding the constraints for standard binary class labels, we can formulate the following optimization problem:

$$\min_{\mathbf{w},w_0,\mathbf{b},\boldsymbol{\eta},\Xi} \frac{1}{2}||\mathbf{w}||_2^2 + B\sum_{i=1}^{N}\eta_i + C\sum_{j=1}^{m-1}\sum_{i=1}^{N}\xi_{j,i}$$

$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 - \eta_i \qquad \forall i$$

$$z_{j,i}(\mathbf{w}^T\mathbf{x}_i - b_j) \geq 1 - \xi_{j,i} \qquad \forall i, j$$

$$\eta_i, \xi_{j,i} \geq 0 \qquad \forall i, j$$

where $j = 1, 2, \ldots, m-1$ indexes bin bounds in $\mathbf{b}$, and $i = 1, 2, \ldots, N$ indexes data entries. The first term in the objective function is the regularization term, the second term (single sum) defines the hinge loss on binary labels, and the third term (double sum) defines the loss function between each pair of bin bound and each Likert-scale label. η_i and $\xi_{j,i}$ are slack variables permitting violations of binary class and Likert-scale bins respectively. B and C are constants weighing the objective function terms. $z_{j,i}$ is an indicator whether the projection of feature vector $\mathbf{x}_i$ is supposed to be greater or less than the bin bound b_j. If $j < u_i$, indicating the projection of $\mathbf{x}_i$ is supposed to be greater than b_j, $z_{j,i} = 1$, otherwise $z_{j,i} = -1$. In this model, the number of constraints is reduced to roughly $M = mN$. Since Likert-scales typically comes as from 2 to 10 ordering categories with

5 or 7 being the most common, we have $m << N$. Considering the $O(M^3)$ complexity of convex quadratic optimization problems, the time complexity is reduced to $O(m^3N^3)$.

4.2.2.1 Removing empty bins One practical concern related to the above optimization problem occurs when the size $|L|$ of the labeled data is small, and some Likert-scale categories are absent in L. Fortunately, this problem has an easy fix. If a Likert-scale category is missing in L, it is not necessary to consider it, and we should only try to enforce ordering constraints among non-empty ordinal categories. Effectively this translates to a smaller number of bins and their boundaries in the optimization problem.

4.2.3 Active learning

The next challenge is to embed the above learning algorithm in a practical active learning framework. The heart of any active learning method is a strategy that is used to select examples to be queried next. In this work, we propose and experiment with a strategy called expected model change (EMC) [Tong and Koller, 2000, Settles et al., 2008b] that evaluates and measures the potential of an unlabeled data instance to change the model by estimating its impact on instance predictions when it is assumed labeled. Our expected model change (EMC) also uses a Bayesian posterior to calculate the expectation. To prevent the re-training of "add-one" models when adding an unlabeled instance and a possible Likert-scale label into the labeled data, which is typically inefficient and required for traditional EMC strategy, we also train the add-one models incrementally from the current model rather than from scratch, which remarkably reduces the time consumption.

4.2.3.1 Expected model change Briefly, the expected model change (EMC) [Tong and Koller, 2000, Settles et al., 2008b] of an unlabeled sample $\mathbf{x}$ can be measured as follows: Suppose that, for the labeled data L, we have already trained a model f_L. For $\mathbf{x}$, there are m possible Likert-scale labels (m is the number of Likert-scale ordinal categories). For each possible Likert-scale label u, if we add $\langle \mathbf{x}, u \rangle$ into L, we will obtain an add-one model $f_{L \cup \langle \mathbf{x}, u \rangle}$. The model change of $f_{L \cup \langle \mathbf{x}, u \rangle}$ compared with f_L is denoted as $\delta(\mathbf{x}, u)$. Since there are m possible

Likert-scale labels, we will have m performance changes $\delta(\mathbf{x}, u)$ where $u = 1, 2, \ldots, m$, each corresponding to one add-one model. The expected model change $\Delta(\mathbf{x})$ of $\mathbf{x}$ is then calculated as:

$$\Delta(\mathbf{x}) = \sum_{u=1}^{m} p(u|\mathbf{x})\delta(\mathbf{x}, u)$$

4.2.3.2 Measuring model change One critical question of the expected model change framework is, how to measure the model change $\delta(\mathbf{x}, u)$ for an unlabeled example $\mathbf{x}$ and one possible label u the example can be assigned to. In this work, we adopt the measurement based on the discrepancy of the predictions over unlabeled data for cases before and after $\mathbf{x}$ and u are added into L and used to learn a new model. More formally, this measurement is calculated as follows: Let the model for L be f_L, and the model after $\langle \mathbf{x}, u \rangle$ is added to L be $f_{L \cup \langle \mathbf{x}, u \rangle}$. For any unlabeled sample $\mathbf{x}_i$, we measure the model change as the discrepancy of the bin predictions $||bin(f_L(\mathbf{x}_i)) - bin(f_{L \cup \langle \mathbf{x}, u \rangle}(\mathbf{x}_i))||$. By considering every unlabeled example, the net model change $\delta(\mathbf{x}, u)$ can be calculated by averaging its impact on all unlabeled data as:

$$\delta(\mathbf{x}, u) = \sum_{i \in U} ||bin(f_L(\mathbf{x}_i)) - bin(f_{L \cup \langle \mathbf{x}, u \rangle}(\mathbf{x}_i))||$$

where i indexes all examples in the unlabeled dataset U.

4.2.3.3 Approximating the expectation After calculating performance changes $\delta(\mathbf{x}, u)$ for all possible ordinal labels u, one important question is how to calculate the expectation needed for the expected model change score. In this work, we adopt a Bayesian method to estimate the expectation.

Our calculation is based on the model f_L learned from the labeled set L of data instances. Briefly, a model f_L together with its bin boundaries defines a model for all ordinal categories. We can use this model and its bins to estimate the empirical distribution of labeled examples in these bins. More specifically, each bin that is associated with the projection f_L may receive (labeled) examples from all categories (that is, even categories that do not match the category corresponding to the bin). Assuming there are m categories, in general, each bin may see examples from m different categories. We can use the observed counts of the examples with these categories that fall

into the same bin to calculate the necessary expectations for an unlabeled data point **x**. Briefly, we take an unlabeled data point and use the projection f_L to identify the bin it falls into. The count of category labels for this bin is then used to approximate their probability distribution and hence calculate the expected value.

More formally, let **x** be an unlabeled instance we are considering to query and $j = bin(f_L(\mathbf{x}))$ be the bin category the instance falls into based on f_L. Our objective is to estimate the probability distribution $(p_1^j, p_2^j, \ldots, p_m^j)$ for bin j, which represents the probability of an example in bin j to be assigned to one of the m Likert-scale categories. One way to estimate this probability would be to use the maximum likelihood approach and calculate the probabilities from counts of Likert-scale labels $q_1^j, q_2^j, \ldots, q_m^j$ in L that fall into bin j, and by assuming they follow a multinomial distribution with parameters $(p_1^j, p_2^j, \ldots, p_m^j)$. However, this estimate may not work well if the number of labeled examples is small which would lead to a biased estimate. Hence, instead of the maximum likelihood based estimate, we base our estimate on the posterior distribution.

To estimate the posterior distribution of $(p_1^j, p_2^j, \ldots, p_m^j)$ we use a Dirichlet prior which is the conjugate choice for the multinomial sampling distribution. Since we do not have any prior information about the distribution of categories in the bin; we choose $Dirichlet(1, 1, \ldots, 1)$ where all Likert-scale categories are assigned the same prior probability. Given the conjugate prior, the posterior of $(p_1^j, p_2^j, \ldots, p_m^j)$ for L follows a Dirichlet distribution:

$$(p_1^j, p_2^j, \ldots, p_m^j)_L \sim Dirichlet(1 + q_1^j, 1 + q_2^j, \ldots, 1 + q_m^j).$$

Given the posterior distribution, we can approximate the probability $p(u|\mathbf{x})$, that is, the probability that **x** is assigned label u, by the expected value of $E(p_u^j)$ from the posterior distribution:

$$E(p_u^j) = \frac{(1 + q_u^j)}{(m + \sum_{i=1}^{m} q_i^j)}.$$

Substituting the result, we can finally calculate the expected model change for an unlabeled sample **x** as:

$$\Delta(\mathbf{x}) = \sum_{u=1}^{m} p(u|\mathbf{x})\delta(\mathbf{x}, u) = \sum_{u=1}^{m} \frac{(1 + q_u^j)}{(m + \sum_{i=1}^{m} q_i^j)}\delta(\mathbf{x}, u)$$

4.2.3.4 Counting to preserve ordering information

One concern of adopting a multinomial distribution to model the data is that all categories in the multinomial model are assumed to be independent. However, our approach uses Likert-scale categories, which are ordinal categories. One way to modify the multinomial model to reflect such dependencies is to use partial counts and let categories close to the category assigned for example $\mathbf{x}_i$ take partial credit for it. To implement this idea we modify the counts $q_1^j, q_2^j, \ldots, q_m^j$ associated bin j as follows: if an observed example $\mathbf{x}_i$ that falls into bin j is assigned a Likert-scale label u_i then it contributes 1 to the count $q_{u_i}^j$ and 0.5 to the counts $q_{u_i-1}^j$ and $q_{u_i+1}^j$ (that is, two Likert-scale categories next to the observed category).

4.2.4 Training of add-one models

Another critical question is the running time complexity to obtain an add-one model $f_{L\cup\langle\mathbf{x},u\rangle}$ after adding an unlabeled sample $\mathbf{x}$ and possible Likert-scale label u into labeled data L. If we train the add-one model from scratch, the time complexity is $O(m^3|L|^3)$ where m is the number of Likert-scale labels. Since, in order to select the sample to be labeled next, we need to obtain an add-one model for each unlabeled sample and each possible Likert-scale label, the total time complexity is $O(m^4|L|^3|U|)$ (U is the unlabeled data), which is extravagant and does not scale well as the size of L grows. To solve this problem, we develop an incremental solver learning classifiers from ordinal category feedback. This solution extends the incremental SVM solver proposed in [Poggio and Cauwenberghs, 2001]. By using the incremental solver when training $f_{L\cup\langle\mathbf{x},u\rangle}$, instead of starting from scratch, we always start from f_L, which remarkably reduces the total time complexity to $O(m^4|L|^2|U|)$.

4.3 Experiments and results

We test our approach on both synthetic and real-world data. The first set of experiments uses data from several UCI regression and ordinal classification datasets which we transform to problems with Likert-scale categories. The second experiment works with real-world clinical data with true (human assessed) ordinal categorical labels.

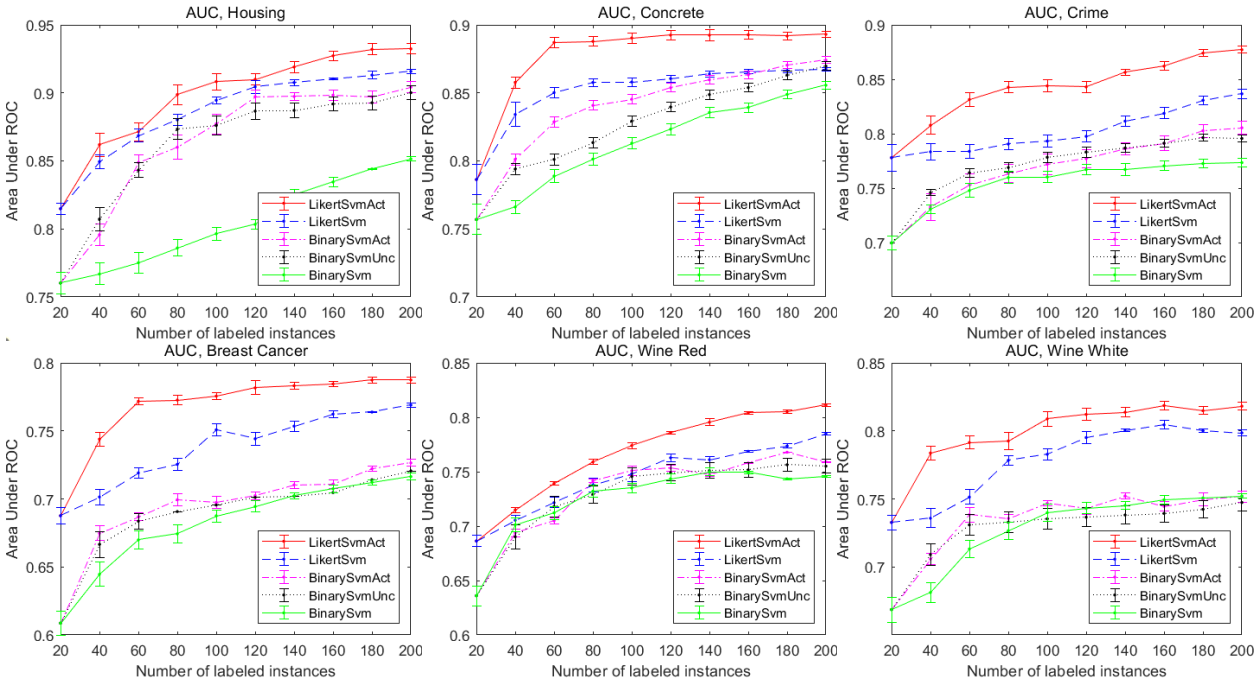

Figure 10: Performance regarding different labeled instance numbers on six synthetic datasets.

4.3.1 Experiments on synthetic UCI-based data

In this part, we adapted three UCI regression datasets (Housing, Concrete, and Crime) and three UCI ordinal classification datasets (Cancer, Wine Red, and Wine White) that are summarized in Table 1 as follows.

For the regression datasets, we discretized the real-valued outputs into 7 Likert-scale levels by dividing the range of output values into equal length bins. We defined a binary class label by considering the examples that fell into three higher-value bins as representatives of class 1 and examples in four lower-value bins as examples from class 0. For example, in Housing dataset this discretization would represent houses with high attractiveness (class 1), and houses with low attractiveness (class 0) and Likert scales represent different degrees of attractiveness. The UCI ordinal classification datasets come with multiple (ordinal) classes so that they can be used as Likert-scale levels directly. The binary thresholds can be set according to the meaning of these ordinal classes. For example, Breast Cancer dataset contains six ordinal classes $\{1, 2, 3, 4, 5, 6\}$,

Dataset	# Samples	# Features	# Categories
Housing	506	13	Regression
Concrete	1030	9	Regression
Crime	1994	122	Regression
Breast Cancer	699	10	6
Wine Red	1599	12	11
Wine White	4898	12	11

Table 1: Properties of all synthetic datasets in experiments.

where $\{1, 2\}$ are healthy, and $\{3, 4, 5, 6\}$ represent the different stages of malignancy, so we map Likert-scale levels 3,4,5,6 to Class 1 and the rest to Class 0.

The objective of our experiments is to demonstrate the benefits of our active learning strategy for models of Likert-scale labels by comparing it to different classification models trained on Likert-scale versus binary labels, and labeling strategies based on the random versus active sampling. Our experiments compare the following models:

BinarySvm: The standard linear SVM with the hinge loss and quadratic regularization factor trained on examples with binary labels only that were sampled randomly.

BinarySvmUnc: The standard linear SVM with the hinge loss and quadratic regularization factor trained on examples with binary labels only, but sampled actively based on the uncertainty sampling selection criterion.

BinarySvmAct: The standard linear SVM with the hinge loss and quadratic regularization factor trained on examples with binary labels only, but sampled actively based on the expected model change (EMC) selection criterion. To apply the criterion to binary classification settings, we treat class 0 and class 1 as two bins.

LikertSvm: Our SVM-based for Likert-scale labels that enforces both binary and bin-label constraints. Examples to be labeled next, are selected randomly.

LikertSvmAct: A combination of our SVM-based for Likert-scale labels and our expected model change for selecting examples to be labeled next.

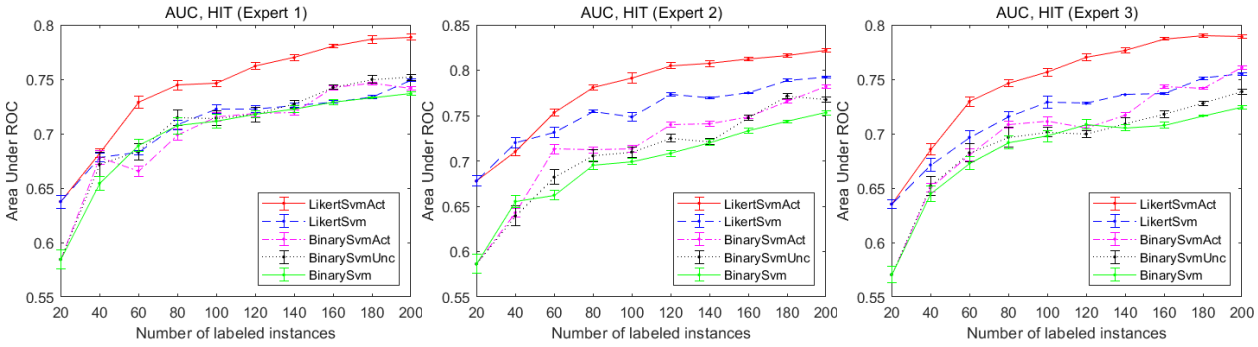

Figure 11: Performance on real-world HIT dataset annotated by three experts.

We evaluated the performance of the different methods by calculating the Area under the ROC (AUC) the learned classification model would achieve on the test data. Hence, each dataset before the learning was split into the training and test set (using $\frac{2}{3}$ and $\frac{1}{3}$ of all data entries respectively). The active learning considered training data only; the AUC was always calculated on the test set. The test set performance reflects how well the model generalizes to future data. To avoid potential train/test split biases, we repeated the training process (splitting) and learning steps 24 times. We report the average AUC obtained on these test sets. To test the benefits of our active learning strategy and the impact of Likert-scale label information on the number of data entries, we trace the performance of all models for the different sizes M of labeled data. Figure 10 shows the performance (AUC) of the models on all six UCI datasets for increasing sizes of M.

Figure 10 shows the benefit of LikertSvmAct with a combination of our active learning strategy and Likert-scale feedback. Both LikertSvmAct and LikertSvm outperform BinarySvmAct, BinarySvmUnc, and BinarySvm, indicating that Likert-scale feedback models will achieve better performance than original binary label models with the same training sizes. LikertSvmAct also outperforms LikertSvm, validating the effectiveness of our querying strategy. Meanwhile, LikertSvmAct greatly outperforms BinarySvm, indicating the combination of active learning and Likert-scale labels clearly raises the performance on the same sizes of training data.

4.3.2 Experiments on clinical data

While the experiments on synthetic datasets appear to support the benefits of our active learning approach based on ordinal Likert-scale labels, it is unclear whether synthetic labels generated for the UCI datasets do not make any unreasonable assumptions and whether good performance also generalizes to "true" feedback provided by humans. In this set of experiments, we test the performance of the methods on a real-world clinical data that were independently reviewed and assessed in terms of Likert-scale feedback by three different experts. The target label concerns clinician's agreement with raising an alert on Heparin-induced thrombocytopenia (HIT), an adverse clinical condition that affects the patient who is treated with heparin for prolonged periods of time. The data and features for the experiment were derived from the PCP database of Electronic records of post-cardiac surgical patients [Hauskrecht et al., 2010, Hauskrecht et al., 2013, Valko and Hauskrecht, 2010]. The clinical data consists of 50 patient state features essential for detection of HIT. The datasets consist of 579, 571, and 573 labeled patient state instances for Expert 1, 2 and 3 (see [Valizadegan et al., 2013]), respectively. The labels include Likert-scale labels on 4 levels indicating the agreement, weak agreement, weak disagreement, and disagreement of the expert with the HIT alert [Nguyen et al., 2013].

Figure 11 shows the AUC performance of the same methods and models as used in the previous section on three expert-annotated HIT datasets. The performance of LikertSvmAct outperforms LikertSvm, BinarySvmAct, BinarySvm on all three datasets, confirming good performance of our method on synthetic data and the benefit of both the Likert-scale labels and active learning for a more efficient training of binary classification models.

4.4 Summary

In this work, we proposed a new framework for learning binary classification models from human feedback that utilizes a refined human feedback expressed in terms of ordinal Likert-scale categories and novel active learning strategy. Our results on synthetic and real-world clinical data show that our learning framework (1) can learn more efficiently and from a smaller number of

examples than existing methods (2) is better than models that rely on Likert-scale labels or active learning individually.

5.0 Active Learning of Multi-class Classification Models from Probabilistic Scores

5.1 Introduction

The work covered in this chapter was accepted and published in the 2018 conference of Florida Artificial Intelligence Research Society (FLAIRS) [Xue and Hauskrecht, 2018]. In this chapter, we explore two strategies for multi-class classification models to alleviate the annotation effort and their combination: probabilistic scores and active learning.

Multi-class classification models are typically learned from annotated data in which every data instance is associated with one class label indicating the class choice assigned to it by a human annotator. In addition to class labels, we may also ask the annotator to provide probabilistic scores in a similar way to Chapter 3, where each data instance is associated with a probabilistic score indicating the certainty of human annotators in the given class label, such as, a probability of the patient having a disease. A more ubiquitous example is the comprehensive evaluation of papers, where the reviewer is asked to provide the strongest advantage of one paper and how strong such advantage is. Here we can treat the strongest advantage as class label and the extent of strongness as probabilistic score. In this chapter, we show how to improve and extend our approach based on ordinal regression and ranking-SVM for binary classification from probabilistic scores in Chapter 3 to multi-class classification settings. The new method is one of the contributions of this chapter. We also develop a new active learning strategy, expected approximate projection change (EAPC), assuming the feedback also includes the probabilistic score in addition to class label. Our active learning strategy implements a variant of the expected model change (EMC) approach. The EMC approach requires costly recalculation of models every time an instance is considered during the example selection process. We address it by developing its efficient gradient-based approximation instead, which remarkably reduces the time consumption.

Through experiments, we show that our new multi-class classification framework achieves improved classification performance and, at the same time, it is able to speed up the selection of instances to be queried next by its active learning component. These results are obtained on both

simulated data derived from data in UCI repository and real-world image data. We demonstrate the ability of our active learning and probabilistic score solutions to reduce the data labeling cost both individually and in combination.

5.2 Methodology

5.2.1 Multi-class support vector machine with probabilistic scores

5.2.1.1 Problem settings Our goal is to learn a multi-class classifier $f : X \to Y$, where X is the feature space and $Y \in \{1, 2, \ldots, k\}$ represents class labels of a data instance. We assume that in addition to class labels $\{1, 2, \ldots, k\}$ defining y_i we also obtain probabilistic scores: a probability p_i reflecting annotator's confidence the example $\mathbf{x}_i$ belongs to class y_i. Hence each labeled data entry D_i consists of three components: $D_i = (\mathbf{x}_i, y_i, p_i)$, an input, a class label and an estimate of the probability of the class label.

5.2.1.2 Learning a multi-class classifier with probabilistic scores To elaborate our multi-class classifier with probabilistic scores, we need to incorporate the probabilistic scores into the multi-class support vector machine (Section 2.2.1). Perhaps the most straightforward intuition is to incorporate the exact probabilistic scores. For example, we may reformulate the k binary classifiers into k regression models based on the probabilistic scores. However, it is well known that humans are often unable to give consistent probabilistic assessments [Juslin et al., 1998, Griffin and Tversky, 1992]. In other words, probabilistic scores from human annotators are usually noisy which may backfire if we dwell too strongly on their exact values. To handle this, we incorporate the probabilistic scores via constraints derived from ordinal regression [Chu and Keerthi, 2005], which was first proposed by [Xue and Hauskrecht, 2017b] for binary classifiers. Briefly, we first split the probabilistic score space into multiple consequent and non-overlapping bins for each one-vs-all classifier. Then we try to enforce the pairwise orderings between each bin boundary and each probabilistic score in this class. Formally, for each one-vs-all classifier $f_j(\cdot)$ and each data instance $\langle \mathbf{x}_i, y_i, p_i \rangle$ such that $y_i = j$, we try to enforce its projection

$f_j(\mathbf{x}_i)$ will fall into the bin consistent with its probabilistic score p_i. Meanwhile, we still try to enforce that $f_j(\mathbf{x}_i)$ is the highest among all one-vs-all classifiers. For example, if a data instance $\mathbf{x}$ belongs to class 3 and probabilistic score 0.4, we want to enforce that the projection f_3 distinguishing class 3 will not only put $\mathbf{x}$ into the bin consistent with its soft label 0.4, but also is greater than any other projection $f_l(\mathbf{x})$ where $l \in \{1, 2, \ldots, k\} \setminus 3$ to guarantee that $\mathbf{x}$ will still be predicted as class 3. Also, we allow violations of both kinds of constraints by penalizing the loss function. By combining two kinds of constraints, we can formulate the following optimization problem:

$$\begin{aligned}
\min_{W,\mathbf{w}_0,H,\Xi,\mathbf{b}} G &= \frac{1}{2}\sum_{l=1}^{k}||\mathbf{w}_l||_2^2 + B\sum_{i=1}^{N}\sum_{l\neq y_i}\eta_{i,l} + C\sum_{i=1}^{N}\sum_{j=1}^{m-1}\xi_{i,j} \\
(\mathbf{w}_{y_i} - \mathbf{w}_l)^T\mathbf{x}_i + (w_{0,y_i} - w_{0,l}) &\geq 1 - \eta_{i,l} \qquad \forall i, l \\
z_{i,j}(\mathbf{w}_{y_i}^T\mathbf{x}_i + w_{0,y_i} - b_j) &\geq 1 - \xi_{i,j} \qquad \forall i, j \\
\eta_{i,l} &\geq 0 \qquad \forall i, l \\
\xi_{i,j} &\geq 0 \qquad \forall i, j
\end{aligned} \tag{5.1}$$

where y_i is the class label of $\mathbf{x}_i$, $z_{i,j}$ is an indicator whether the projection of $\mathbf{w}_{y_i}^T\mathbf{x}_i$ is supposed to be greater or less than the jth bin boundary b_j (-1 for less and 1 for greater); $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k$ and $w_{0,1}, w_{0,2}, \ldots, w_{0,k}$ are the parameters and biases for the k binary classifiers.

5.2.2 Active learning

In this part, we develop an active learning framework that builds a multi-class classifier by actively querying a human annotator for assessing the instances using both the class and associated probabilistic scores. We show how this algorithm can be included in the active learning framework that aims to improve the model by wisely selecting the examples to be assessed next. The criterion used to choose from among unlabeled candidate instances is based on the highest expected approximate projection change.

5.2.2.1 Expected approximate projection change In this chapter, the expected approximate projection change (EAPC) is inspired by the expected model change [Tong and Koller, 2000]. Briefly, expected approximate projection change selects the unlabeled instance that brings the greatest expected projection change when it is assumed labeled. Such strategy consists of two key quantities: projection change and expectation. When an unlabeled instance is assigned an assumed label, all the k one-vs-all classifiers will change, leading to changes in projections of all unlabeled instances. The projection change measures the absolute change of all unlabeled instances on all the k one-vs-all classifiers. Since in multi-class classification scenario with probabilistic scores, an assumed label contains a discrete class label and a continuous probabilistic score, given the probability of each class label and conditional distribution of the probabilistic score, we can calculate the expectation of projection change over the space of assumed label for the unlabeled instance. Formally, when an unlabeled instance $\mathbf{x}^+$ is assigned an assumed label $\langle y^+, p^+ \rangle$, the current models $f_{i,L}(\cdot)$ built on labeled data L will change to $f_{i,L\cup\langle \mathbf{x}^+, y^+, p^+\rangle}(\cdot)$ for all i. Given the probability $P(y^+|\mathbf{x}^+)$ and conditional density $p(p^+|\mathbf{x}^+, y^+)$, we can calculate the expected projection change $\Delta(\mathbf{x}^+)$ as:

$$\Delta(\mathbf{x}^+) = \sum_{y^+}(y^+|\mathbf{x}^+) \int_0^1 p(p^+|\mathbf{x}^+, y^+) \sum_{i=1}^{k} \sum_{j\in U} |f_{i,L\cup\langle \mathbf{x}^+, y^+, p^+\rangle}(\mathbf{x}_j) - f_{i,L}(\mathbf{x}_j)| dp^+$$

We select the unlabeled instance with highest expected projection change to be labeled next. To prevent the re-training of "add-one" models when adding an unlabeled instance, a class label and a possible probabilistic into the labeled data, which is typically inefficient and required for traditional EMC strategy, we also propose an approximation based on the gradient of the add-one models instead of training them, which remarkably reduces the time consumption.

5.2.2.2 Approximating expectation One critical problem is the expectation. Unfortunately, since the probabilistic score space is continuous, it is typically unfeasible to obtain the probabilistic score distribution of an unlabeled instance directly. To solve this problem, we propose an approximation which splits the probabilistic score range into multiple consequent and non-overlapping segments, then calculate the conditional probability that the unlabeled instance falls into each segment. Since such approximation is similar to the binning strategy in Equation 5.1 we can directly adopt the bins for the conditional probabilities. Formally, instead of conditional

density, we split the probabilistic score range into m bins $\{q_1, q_2, \ldots, q_m\}$ and calculate the conditional probability $P(p^+ \in q^+|\mathbf{x}^+, y^+)$ for all i and q^+. Therefore, the expectation $\Delta(\mathbf{x}^+)$ can now be estimated as:

$$\Delta(\mathbf{x}^+) = \sum_{y^+}(y^+|\mathbf{x}^+) \sum_{q^+} P(p^+ \in q^+|\mathbf{x}^+, y^+) \sum_{i=1}^{k} \sum_{j \in U} |f_{i,L \cup \langle \mathbf{x}^+, y^+, p^+ \in q^+ \rangle}(\mathbf{x}_j) - f_{i,L}(\mathbf{x}_j)|$$

Another problem is measurement of $P(y^+|\mathbf{x}^+)$ and $P(p^+ \in q^+|\mathbf{x}^+, y^+)$. In this work, we adopt the idea of density weight [Settles et al., 2008a]. Briefly, if an unlabeled instance is closed to a labeled instance, they are of high probability with the identical label. Formally, for an unlabeled instance $\mathbf{x}^+$ and each labeled instance $\langle \mathbf{x}_i, y_i, p_i \rangle$, the probability they are with identical label is proportional to the inverse of their Euclidean distance $||\mathbf{x}_i - \mathbf{x}^+||_2$. Therefore, the joint probability $P(y^+|\mathbf{x}^+)P(p^+ \in q^+|\mathbf{x}^+, y^+) = P(y^+, p^+ \in q^+|\mathbf{x}^+)$ can be estimated as:

$$P(y^+|\mathbf{x}^+)P(p^+ \in q^+|\mathbf{x}^+, y^+) = \frac{1}{Z} \sum_{i \in L}^{y_i = y^+, p_i = p^+} \frac{1}{||\mathbf{x}_i - \mathbf{x}^+||_2}$$

where $Z = \sum_{i \in L} \frac{1}{||\mathbf{x}_i - \mathbf{x}^+||_2}$ is the normalization factor.

5.2.2.3 Approximating projection change Another concern is the projection change over the unlabeled data. When adding an unlabeled instance with an assumed label, the new "add-one" model should be retrained. Given U unlabeled instances, k classes, m bins (in Equation 5.1, probabilistic scores in the same bin give the identical optimization), we need to retrain kmU "add-one" models. To avoid retraining, we propose an approximation via gradient inspired by stochastic gradient descent [Bottou and Bousquet, 2008]. Briefly, when adding an unlabeled instance with an assumed label, we can treat the other (labeled) instances as constants, calculate the difference compared with the current model and take the gradient to approximate the projection change over the unlabeled data. Formally, when adding $\langle \mathbf{x}^+, y^+, p^+ \in q_j \rangle$ into Equation 5.1, the new "add-one" model G^+ can be written via rectified function $[\cdot]_+$ (we omit the bias $w_{0,l}$ for convenience) as:

$$\min_{W, \mathbf{w}_0, H, \Xi, \mathbf{b}} G^+ = \frac{\mathbf{w}^T\mathbf{w}}{2} + B \sum_{l \neq y^+} [1 - (\mathbf{w}_{y^+} - \mathbf{w}_l)^T \mathbf{x}^+]_+ + B \sum_{i=1}^{N} \sum_{l \neq y_i} [1 - (\mathbf{w}_{y_i} - \mathbf{w}_l)^T \mathbf{x}_i]_+ + C \sum_{j=1}^{m-1} [1 - z_j^+ (\mathbf{w}_{y^+}^T \mathbf{x}^+ - b_j)]_+ + C \sum_{j=1}^{m-1} \sum_{i=1}^{N} [1 - z_{i,j} (\mathbf{w}_{y_i}^T \mathbf{x}_i - b_j)]_+$$

where z_j^+ is determined from p^+ for all j. Comparing with Equation 5.1, we get:

$$\Delta G^+ = B \sum_{l \neq y^+} [1 - (\mathbf{w}_{y^+} - \mathbf{w}_l)^T \mathbf{x}^+]_+ + C \sum_{j=1}^{m-1} [1 - z_j^+ (\mathbf{w}_{y^+}^T \mathbf{x}^+ - b_j)]_+$$

Therefore, the gradient for each one-vs-all classifier can be calculated as:

$$\frac{\partial \Delta G^+}{\partial \mathbf{w}_l} = B\mathbf{x}^+ \mathbb{1}_{(\mathbf{w}_{y^+} - \mathbf{w}_l)^T \mathbf{x}^+ < 1} \quad (l \neq y^+)$$

$$\frac{\partial \Delta G^+}{\partial \mathbf{w}_{y^+}} = -B\mathbf{x}^+ \sum_{l \neq y^+} \mathbb{1}_{(\mathbf{w}_{y^+} - \mathbf{w}_l)^T \mathbf{x}^+ < 1} - C\mathbf{x}^+ \sum_{j=1}^{m-1} z_j^+ \mathbb{1}_{z_j^+ (\mathbf{w}_{y^+}^T \mathbf{x}^+ - b_j) < 1}$$

In the stochastic gradient descent, the negative gradient determines the step length for learning. Therefore, we claim the gradient is approximately proportional to the change of the parameter of each one-vs-all classifier:

$$\Delta \mathbf{w}_l^+ \propto \frac{\partial \Delta G^+}{\partial \mathbf{w}_l} \qquad l = 1, 2, \ldots, k$$

Given an arbitrary unlabeled instance $\mathbf{x}_j$, we can approximate the absolute projection change on $\mathbf{w}_i$ before and after $\langle \mathbf{x}^+, y^+, p^+ \in q^+ \rangle$ as:

$$|f_{i, L \cup \langle \mathbf{x}^+, y^+, p^+ \in q^+ \rangle}(\mathbf{x}_j) - f_{i,L}(\mathbf{x}_j)| = |\frac{\partial \Delta G^+}{\partial \mathbf{w}_i}^T \mathbf{x}_j|$$

5.3 Experiments and results

We test our framework on both synthetic and real-world data. The first experiment adapts data from two UCI multi-class data sets which we transform to multi-class classification tasks with probabilistic scores. The second experiment works with a real-world image data with human assessed labels from multiple annotators.

5.3.1 Experiments on simulated data

5.3.1.1 Data simulation We adapted two UCI multi-class datasets (see Table 2 for details) as follows: We take half of the data to train a multi-class support vector machine and obtain probabilistic scores on the other half via soft-max function on their predictions. In the experiments we use only the second half of the data, retain the class labels and keep the corresponding probabilistic scores.

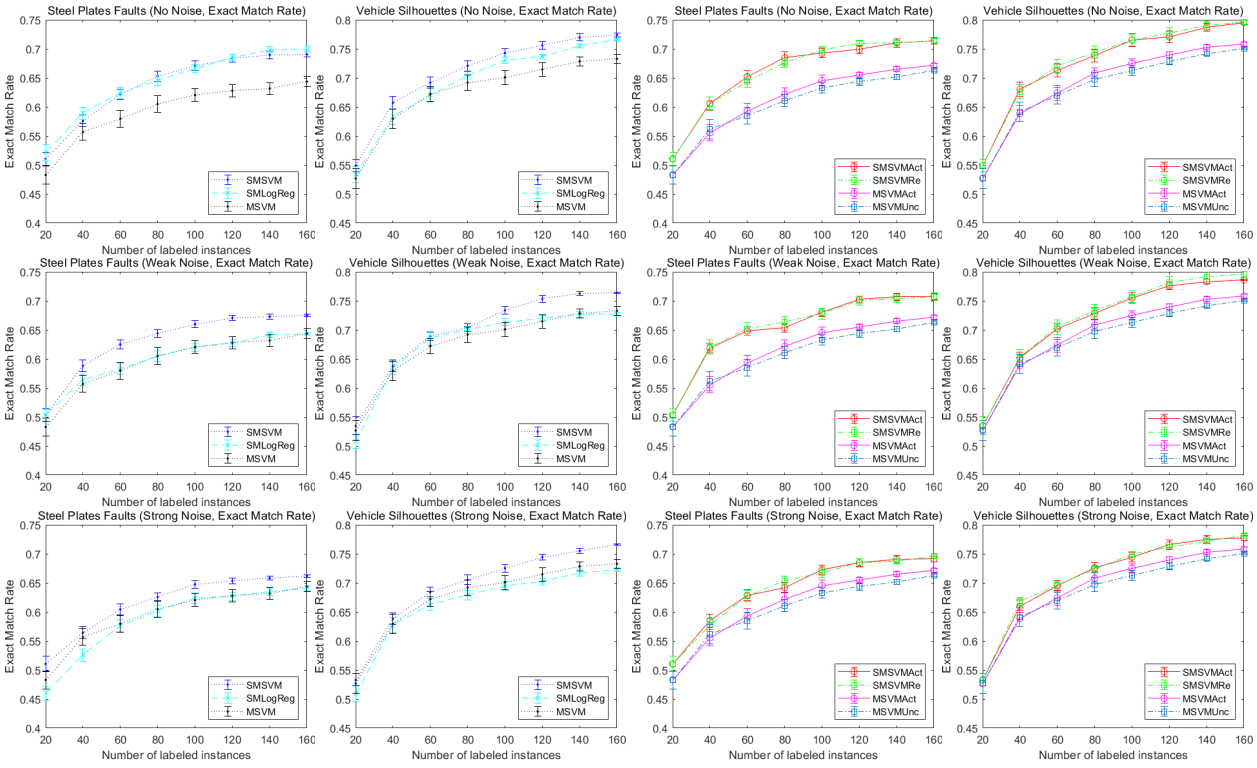

Figure 12: Performance (EMR) on two synthetic datasets regarding different labeled instance numbers with no (top), weak (middle) and strong (bottom) noise.

5.3.1.2 Experimental settings To demonstrate the benefits of our model with probabilistic scores and expected approximate projection change strategy, we compare it with multi-class classifiers trained only on class labels, multi-class logistic regression with probabilistic scores and active learning that retrains to calculate the exact projection change when adding an unlabeled instance. Our experiments compare the following classifiers (we use random sampling by default):

MSVM: multi-class support vector machine [Vapnik, 1998, Weston et al., 1999] where K one-vs-all classifiers are trained jointly;

MSVMUnc: multi-class support vector machine [Vapnik, 1998, Weston et al., 1999] where K one-vs-all classifiers are trained jointly with uncertainty sampling;

MSVMAct: multi-class support vector machine [Vapnik, 1998, Weston et al., 1999] where K one-vs-all classifiers are trained jointly with expected approximate projection change strategy;

SMLogReg: multi-class logistic regression with probabilistic scores where K one-vs-all classifiers are trained independently on exact probabilistic scores;

SMSVM: multi-class support vector machine with probabilistic scores where K one-vs-all classifiers are trained jointly with probabilistic score constraints;

SMSVMRe: multi-class support vector machine with probabilistic scores where K one-vs-all classifiers are trained jointly with probabilistic score constraints and retraining the model to calculate the exact projection change when an unlabeled instance is added;

SMSVMAct: multi-class support vector machine with probabilistic scores where K one-vs-all classifiers are trained jointly with probabilistic score constraints and expected approximate projection change strategy.

We evaluate the performance of the different methods in the exact match rates (EMR) on the test data. All data sets before learning are split into the training and test set (using $\frac{2}{3}$ and $\frac{1}{3}$ of all data instances). The learning considered training data only; the EMR is always measured in the test set. We also repeat the splitting and learning 24 times. The average EMRs of different classifiers on UCI data regarding increasing sizes of N are reported in Figure 12.

5.3.1.3 Experimental results Figure 12 (top) shows the benefit of our framework *SMSVMAct* with a combination of probabilistic scores and expected approximate projection change strategy. Both *SMSVMAct* and *SMSVMRe* outperform *MSVMAct* and *MSVMUnc*; both *SMSVM* and *SMLogReg* outperform *MSVM*. These two comparisons show probabilistic scores will achieve better performance than original class label models with the same training sizes. Meanwhile, both *SMSVMAct* and *SMSVMRe* outperform *SMSVM*; *MSVMAct* outperforms *MSVMUnc* and *MSVM*. These two comparisons show the effectiveness of our expected approximate projection change strategy. Overall, both *SMSVMAct* and *SMSVMRe* are always of highest performance, showing

that our framework remarkably raises the performance on the same sizes of training data.

5.3.1.4 Noise simulation In order to generate noise in probabilistic scores, each probabilistic score p derived from the UCI data was modified into p' by injecting a Gaussian noise of different strength:

Weak noise: $p' = p \times (1 + 0.1 \times N(0, 1))$;

Strong noise: $p' = p \times (1 + 0.3 \times N(0, 1))$.

Briefly, the noise injection levels above indicate the average proportion of noise to no, weak (10%) and strong (30%) levels respectively. Also, we truncated the illegal probabilistic scores (e.g., probabilistic scores that are less than $\frac{1}{k}$ or greater than 1) to the interval of $[\frac{1}{k}, 1]$.

Dataset	# Instances	# Features	# Classes
Steel Plates Faults	1941	27	7
Vehicle Silhouettes	946	18	4

Table 2: Properties of two synthetic datasets in experiments.

5.3.1.5 Experimental results with noise When noise is added into probabilistic scores, the performance of probabilistic score models may deteriorate. Figure 12 (middle) and (bottom) shows the robustness of our framework *SMSVMAct*. The regression based model *SMLogReg*, which is trained on exact probabilistic scores, is vulnerable to noise and deteriorates remarkably. While other probabilistic score models are more robust and do not suffer from much performance drop. Our framework *SMSVMAct* are still of top two performance comparable with *SMSVMRe*, showing the robustness of our framework.

5.3.1.6 Experiments on time consumption The reason we use gradient to approximate projection change is to reduce time consumption. Figure 13 shows the time consumption of three multi-class classifiers with probabilistic scores in experiments on UCI data sets for increasing sizes of training data.

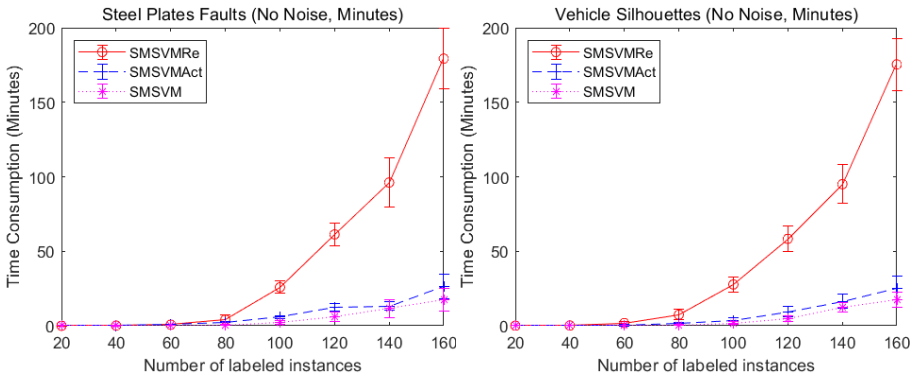

Figure 13: Time consumption (minutes) on synthetic datasets regarding different labeled instance numbers with no noise.

We evaluated the time consumption of the different learning methods by the total minutes elapsed on the training data. Because of the calculation of projection change, both *SMSVMAct* and *SMSVMRe* spend more time than *SMSVM*. However, the time consumption of *SMSVMAct* is tolerable, while the time consumption of *SMSVMRe* is seven times as *SMSVMAct* and ten times as *SMSVM*. Overall, our framework *SMSVMAct*, which combines probabilistic scores and active learning, is of both higher performance than other models that utilize at most one of the two methods, and far more satisfactory time consumption since it prevents retraining.

5.3.2 Experiments on real-world data

We also run experiments on Face Sentiment data, a real-world crowd-sourced dataset from Tsinghua University.

5.3.2.1 Experimental settings Face Sentiment data contains 584 data instances, where each instance is a 128×120 gray-scale photo of the facial expression. The class label is one of the four moods indicating the mood in the photo. Each data instance is annotated by nine annotators. The true label of each data instance is also given. We use a convolutional neural network to extract 256 features for each data instance. For models with probabilistic scores, we take the vote ratio of the true class among nine annotators as the probabilistic score. We split all data instances into $\frac{2}{3}$

training and $\frac{1}{3}$ testing data, and measure average exact match rate over 24 trials.

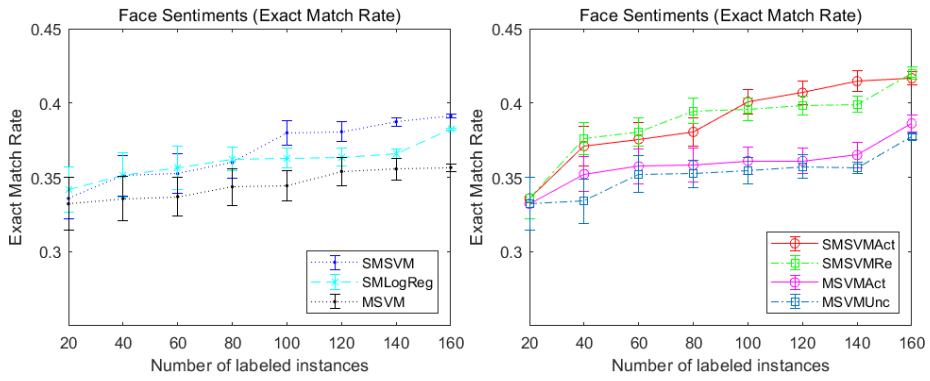

Figure 14: Performance (EMR) of real-world Fact Sentiment data regarding different labeled instance numbers.

5.3.2.2 Experimental results Figure 14 shows the benefit of our framework *SMSVMAct* with a combination of probabilistic scores and expected approximate projection change strategy on real-world face sentiment data. Both *SMSVMAct* and *SMSVMRe* outperform *MSVMAct* and *MSVMUnc*; both *SMSVM* and *SMLogReg* outperform *MSVM*. These two comparisons show probabilistic scores will achieve better performance than original class label models with the same training sizes. Meanwhile, both *SMSVMAct* and *SMSVMRe* outperform *SMSVM*; *MSVMAct* outperforms *MSVMUnc* and *MSVM*. These two comparisons show the effectiveness of our expected approximate projection change strategy. Overall, both *SMSVMAct* and *SMSVMRe* are always of highest performance, showing that our framework remarkably raises the performance on both simulated and real-world data.

5.4 Summary

In this work, we proposed a new framework for multi-class classification models incorporating probabilistic scores and a novel active learning strategy with efficient approximation that: (1) can learn more efficiently and from a smaller number of examples than existing methods, (2)

is of higher performance than models that rely on only probabilistic scores or active learning individually, and (3) can highly reduce time consumption than active learning strategy that requires retraining when adding new data instances.

6.0 Active Learning of Multi-class Classification Models from Ordered Class Sets

6.1 Introduction

The work covered in this chapter was accepted and published in the 2019 AAAI Conference on Artificial Intelligence (AAAI) [Xue and Hauskrecht, 2019]. In this chapter, we explore two strategies for multi-class classification models to alleviate the annotation effort and their combination: ordered class sets and active learning.

Multi-class classification models are typically learned from annotated data in which every data instance is associated with one class label indicating the top class choice assigned to it by a human annotator. However, human annotators can often express and provide additional information about the top class and its relation to other class choices. For example, when the instance is not a clearcut case, there are other likely class choices the annotator may have in mind. Associating multiple competing classes with one instance is common in various diagnostic tasks. For example, in medical domain, a list of competing diagnostic classes is referred to as a differential diagnosis. Briefly, given the features (symptoms, observations, etc.) of a patient, the physician considers not only the leading diagnosis (class), but also other alternative diagnoses (classes) that are possible and may fit the patient's case. Another more ubiquitous example is part-of-speech tagging, where one word is associated with a grammatical tag. Here the annotator can also provide some alternative grammatical tags which, although not as possible as the first grammatical tag, but are also possible choices of this word. The gist of our approach is to utilize such information to learn multi-class classifiers. More specifically, apart from the top class label for each data instance, we let the annotator provide also information about other alternative classes, and express these in terms of the ordered set of classes representing the descending priorities (or confidence) in these classes.

To translate this idea into a working framework we first develop and present a new max-margin multi-class classifier learning method that allows us incorporate the ordered class set (OCS) information into the model learning process. We also explore active learning strategies to further

reduce the annotation effort. We develop a new variant of expected model change (EMC) active learning strategy that considers ordered class set feedback. Specifically, our active learning strategy calculates the expected prediction change for an unlabeled instance by calculating and combining the estimate of the prediction change for the different class-order sets one can assign to the instance. Since, in traditional EMC strategy, the estimate of the expected change requires one to repeat the retraining by considering each unlabeled instance, we propose new approximation strategies that subsamples from the exponential number of possible OCS's for each unlabeled instance and can reduce the running time of the estimate instead. To prevent the re-training of "add-one" models when adding an unlabeled instance and a possible Likert-scale label into the labeled data, which is typically inefficient and required for traditional EMC strategy, we also train the "add-one" models (models with an unlabeled instance and a possible OCS added into the labeled data) incrementally from the current model rather than from scratch, which further reduces the time consumption.

We experiment with our new framework on both synthetic and real-world datasets with class-order feedback. We show the effectiveness of the ordered class set feedback and active learning for reducing the annotation effort both individually and jointly. We also show that our solution outperforms existing multi-class classification methods that consider one-class-per-example labels.

6.2 Methodology

In this part, we develop an active learning framework that builds a multi-class classification model by actively querying an annotator who provides feedback to the framework by assessing instances with OCS. We start by first defining and formalizing the problem of learning from OCS in a multi-class settings. After that, we present an algorithm for learning the multi-class classification model from such feedback. Second, we show how this algorithm can be included in the active learning framework that aims to improve the model by wisely selecting the examples to be assessed next. The criterion used to choose from among unlabeled candidate instances is based on the highest expected change in OCS. Since the calculation of the expected change in OCS is nontrivial, we present our solutions to the following problems: (1) how to model the distribution of OCS for

calculating the expected change, and (2) how to speed up training via incremental solver when adding one unlabeled instance and an OCS.

6.2.1 Multi-class classifier with ordered class sets (OCS)

6.2.1.1 Problem Our objective is to learn a multi-class classifier $f : X \to Y$, where $X \in \mathbb{R}^d$ is the input space and $Y = \{1, 2, \ldots, K\}$ represents possible (mutually exclusive) classes one can assign to an example. Standard way to learn such a model is to use input-output pairs $\langle \mathbf{x}_i, y_i \rangle$. In this work we learn from the input-OCS pairs $\langle \mathbf{x}_i, S_i \rangle$, where the input $\mathbf{x}_i$ is associated with the ordered class set (or OCS) S_i reflecting the annotator's class preferences. The ordered class set S_i is formed by a non-empty subset of classes defining Y. Please note that the information in the input-OCS pair subsumes the information provided in the standard input-output data form. Briefly, we assume $y_i = S_{i1}$, that is, the class label y_i is identical to the first class in ordered class set S_i. In general S_i may contain any number of classes: an ordered set of only one class only indicates the annotator's top class choice; an ordered set of all K classes indicates the annotator provides the complete ordering of all alternative classes. For example, in a 4-class setting, an OCS $\langle 3, 2 \rangle$ indicates this data instance most probably belongs to class 3, then class 2 and is not likely to belong to any other class. Since the class label is identical to the first class in the OCS, the output (class label) of this instance should be 3.

6.2.1.2 AMSVM with ordered class sets (OCS) Now we show how we can combine approximate support vector machine (AMSVM, Section 2.2.2) with ordered class set (OCS). To achieve this, we incorporate the OCS via constraints derived from the ordinal regression [Chu and Keerthi, 2005]. The gist of the approach is that, for every data instance, we split the classes in the its OCS into two subsets: a "higher" subset and a "lower" subset. Each class in the "lower" subset must satisfy one of the two conditions: (1) it is not included in the OCS, or (2) in the OCS, it comes after all the classes from the higher subset. In other words, each class in the "higher" subset should have higher priority than all the classes from the "lower" subset in their projections. If such condition is guaranteed, we may enforce that the average projection of the "higher" subset is higher than the average projection of the "lower" subset. Formally, for

every labeled instance $\langle \mathbf{x}_i, S_i \rangle$ and $j \in \{1, 2, \ldots, |S_i|\}$, the "higher" subset can be constructed as $\{S_{i1}, S_{i2}, \ldots, S_{ij}\}$, where S_{ij} indicates the jth class in S_i, and the "lower" subset consists of all other classes. Then the goal is to try to enforce the average projection $\frac{1}{j}\sum_{a\in\{S_{i1},S_{i2},\ldots,S_{ij}\}} f_a(\mathbf{x}_i)$ of the "higher" subset should be greater than the average projection $\frac{1}{k-j}\sum_{b\notin\{S_{i1},S_{i2},\ldots,S_{ij}\}} f_b(\mathbf{x}_i)$ of the "lower" subset. Therefore, the optimization of AMSVM with OCS can be formulated as:

$$\min_{W,\Xi} \frac{1}{2}\sum_{l=1}^{k} ||\mathbf{w}_l||_2^2 + C\sum_{i=1}^{N}\sum_{j=1}^{|S_i|}\xi_{ij}$$

$$(\frac{1}{j}\sum_{a\in\{S_{i1},\ldots,S_{ij}\}} \mathbf{w}_a - \frac{1}{k-j}\sum_{b\notin\{S_{i1},\ldots,S_{ij}\}} \mathbf{w}_b)^T\phi(\mathbf{x}_i) \geq 1-\xi_{ij} \qquad \forall i,j$$

$$\xi_{ij} \geq 0 \qquad \forall i,j; \tag{6.1}$$

where S_i is the OCS of $\mathbf{x}_i$ and $\phi(\cdot)$ is the projection of kernel space. $W = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$ are parameters of the k binary one-vs-rest classifiers. N is the number of labeled instances. $\Xi = \{\xi_{ij}\}$ for all $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, |S_i|$ index the slack variables for each constraint. For prediction, the class with the highest projection value is selected as the predicted class. Please notice the prediction from our AMSVM with OCS is still a class label.

6.2.2 Active learning with OCS

The next challenge is to embed the above multi-class classifier with OCS in a compatible active learning strategy. The core of any active learning strategy is a schema to select examples to be queried next. In this work, we propose and experiment with a strategy called expected model change [Tong and Koller, 2000, Settles et al., 2008b] that measures the potential of an unlabeled data instance to change the model by estimating its impact on predictions. In this section, we first show how the expected model change of an unlabeled instance can be calculated by considering all OCS of this instance. After that we tackle two related problems: (1) how to obtain the probability of a specific OCS, and (2) how to measure the change of the model given an unlabeled instance and one of its OCS.

6.2.2.1 Expected model change (EMC) Let $\mathbf{f}_L$ denotes a multi-class classifier trained on all currently labeled data. Our objective is to assess how much impact the annotation of a currently unlabeled example $\mathbf{x}_0$ with an OCS can make. Let $\Delta(\mathbf{f}_L, \mathbf{x}_0)$ be a measure of this impact. In this work, we assess the impact in terms of the expected model change and an unlabeled instance with the highest expected model change is selected for the labeling first. We define the expected model change for the OCS feedback as:

$$\Delta(\mathbf{f}_L, \mathbf{x}_0) = \sum_{S_0 \in \mathbf{S}} P(S_0|\mathbf{x}_0)\delta(\mathbf{f}_L, \mathbf{f}_{L\cup\langle \mathbf{x}_0, S_0\rangle}) \tag{6.2}$$

where $\delta(\mathbf{f}_L, \mathbf{f}_{L\cup\langle \mathbf{x}_0, S_0\rangle})$ denotes a model change induced by assigning an ordered class set (OCS) S_0 to example $\mathbf{x}_0$. Intuitively, the expected change is a weighted average of model changes for all possible ordered class sets $\mathbf{S}$ where the weight is a probability of the instance $\mathbf{x}_0$ being assigned an OCS S_0. To simplify the model of $P(S_0|\mathbf{x}_0)$ and its construction we express it in terms of two probabilities: $P(S_0|\mathbf{x}_0) = P(S_0|A_0, \mathbf{x}_0)P(A_0|\mathbf{x}_0)$, where $P(A_0|\mathbf{x}_0)$ is the probability of an unordered class-set A_0 defining S_0, and $P(S_0|A_0, \mathbf{x}_0)$ is the probability of the specific class-order for a fixed A_0. In order to calculate the expected model change three quantities defining it need to be estimated: (1) the probability $P(A_0|\mathbf{x}_0)$ of each unordered class set A_0, (2) the conditional probability $P(S_0|A_0, \mathbf{x}_0)$ for each OCS S_0 given its corresponding unordered class set A_0, and (3) the model change $\delta(\mathbf{f}_L, \mathbf{f}_{L\cup\langle \mathbf{x}_0, S_0\rangle})$. To prevent enumerating all the OCS's when calculating the OCS distribution, we propose an approximation subsampling over all the OCS's. To prevent the re-training of "add-one" models when adding an unlabeled instance and a possible OCS into the labeled data, which is typically inefficient and required for traditional EMC strategy, we train the add-one models incrementally from the current model instead of from scratch. These two techniques remarkably reduce the time consumption. We present details of our solutions to these next.

6.2.2.2 Estimating the probability of an unordered class set The first quantity to be estimated is the probability $P(A_0|\mathbf{x}_0)$ for each unordered class set A_0. We approximate this quantity with the help of an auxiliary multi-label logistic regression model $\mathbf{g}_L$ we train on the data annotated with OCS. The model $\mathbf{g}_L$ maps instances to a class vector of size k indicating whether a class should be included in the unordered class set or not. We define the output of a

multi-label classifier as $\mathbf{z}_i = \mathbf{g}_L(\mathbf{x}_i) = M^T\phi(\mathbf{x}_i)$. The input $\mathbf{x}_i$ of this model is a d-dimensional feature vector of a data instance, and the output $\mathbf{z}_i$ is a class vector of size k indicating whether a class should be included in the unordered class set or not. M is a $d \times k$ matrix of parameters of this model, and $\phi(\cdot)$ is the projection of the kernel space. The training of $\mathbf{g}_L$ is also intuitive: an OCS can be converted into a class vector naturally. If a class is included in the OCS, then the corresponding scalar of this class in the class vector is 1, otherwise the scalar is -1. After converting the OCS of each labeled instance, we will take the feature vector and class vector of each labeled instance for training. In this chapter, we use an improved multi-label logistic regression model by [Xu et al., 2018]. Basically, this multi-label logistic regression considers the topological information of the feature space: the data instances close to each other are more likely to share the same class vector. Formally, the optimization of the model parameter M can be formalized as follows:

$$\min_M \sum_{i=1}^{N} ||M^T\phi(\mathbf{x}_i) - \mathbf{z}_i||^2 + \lambda \sum_{ij} t_{ij}||M^T[\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)]||^2 \tag{6.3}$$

where $\mathbf{x}_i$ and $\mathbf{z}_i$ are the feature vector and class vector. t_{ij} is the topological information between $\mathbf{x}_i$ and $\mathbf{x}_j$. $t_{ij} = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2})$ if $\mathbf{x}_j$ is among the nearest neighbors of $\mathbf{x}_i$, and $t_{i,j} = 0$ otherwise. λ is the parameter trading off the two terms. The number of nearest neighbors is also tunable. After obtaining the optimal parameter $\hat{M}$ from the optimization, the estimate of $\hat{A}_0$ of A_0 can be obtained from the predicted class vector $\hat{\mathbf{z}_0} = \hat{M}^T\phi(\mathbf{x}_0)$.

6.2.2.3 Estimating the conditional probability of an OCS The second quantity to be estimated when calculating the expected model change is the conditional probability $P(S_0|A_0, \mathbf{x}_0)$ of each OCS S_0 given the corresponding unordered class set A_0 and an unlabeled instance $\mathbf{x}_0$. Although it is hard for us to directly estimate $P(S_0|A_0, \mathbf{x}_0)$, the class-wise conditional probability $P(c|A_0, \mathbf{x}_0)$ of a single class $c \in A_0$ can be estimated directly by applying a soft-max function: $P(c|A_0, \mathbf{x}_0) = \frac{\exp(\mathbf{w}_c^T\phi(\mathbf{x}_0))}{\sum_{i \in A_0}\exp(\mathbf{w}_i^T\phi(\mathbf{x}_0))}$. Since each OCS $S_0 \sim A_0$ is a permutation of the unordered class set A_0, the conditional probability $P(S_0|A_0, \mathbf{x}_0)$ can be constructed from the class-wise conditional probability $P(c|A_0, \mathbf{x}_0)$ for all $c \in A_0$ the same way we construct the probability of a permutation.

Formally, the probability $P(S_0|A_0, \mathbf{x}_0)$ can be constructed as:

$$P(S_0|A_0, \mathbf{x}_0) = \prod_{i=1}^{|S_0|} \frac{P(S_{0i}|A_0, \mathbf{x}_0)}{1 - \sum_{j=1}^{i-1} P(S_{0j}|A_0, \mathbf{x}_0)} \tag{6.4}$$

where S_{0i} indicates the ith probable class in S_0.

It seems the conditional probability $P(S_0|A_0, \mathbf{x}_0)$ is perfectly calculated. However, there is an inevitable fact: each S_0 is a permutation of its corresponding unordered class set A_0. This indicates that, given an unordered class set A_0, the number of OCS S_0 such that $S_0 \sim A_0$ is actually $|A_0|!$. Clearly, it is intractable to calculate the conditional probability $P(S_0|A_0, \mathbf{x}_0)$ for all the OCS $S_0 \sim A_0$. To reduce the number of OCS to enumerate, a straightforward method is to do random sub-sampling over all the OCS $S_0 \sim A_0$. However, such a method introduces another problem: is such sub-sample a "good" approximation over all the OCS $S_0 \sim A_0$? That is, is the EMC obtained using this sub-sample similar the EMC obtained by considering all OCS $S_0 \sim A_0$? To solve this problem, we propose the following sub-sampling scheme: first, we create two random sub-samples T_0' and T_0'' over all the OCS $S_0 \sim A_0$ such that: (1) $S_0 \in T_0' \Rightarrow S_0 \sim A_0$ and $S_0 \in T_0'' \Rightarrow S_0 \sim A_0$. In other words, both T_0' and T_0'' only contains the OCS whose corresponding unordered class set is A_0. (2) $T_0' \cap T_0'' = \emptyset$, and (3) $|T_0'| = |T_0''| = m$ where m is a small number. Then, we define an instance-wise EMC $\kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0')$ of the unlabeled instance $\mathbf{x}_0$ on an arbitrary unlabeled instance $\mathbf{x}_u$ and a sub-sample set T_0' where $S_0 \in T_0' \Rightarrow S_0 \sim A_0$ as follows:

$$\kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0') = \frac{1}{Z} \sum_{S_0 \in T_0'} P(S_0|A_0, \mathbf{x}_0) \delta'(\mathbf{f}_L, \mathbf{f}_{L \cup \langle \mathbf{x}_0, S_0 \rangle}, \mathbf{x}_u) \tag{6.5}$$

where $u \notin L$ and T_0' only contains the OCS whose corresponding unordered class set is A_0. $Z = \sum_{S_0 \in T_0'} P(S_0|A_0, \mathbf{x}_0)$ is the partition function. δ' reflects the OCS change observed on a specific unlabeled example $\mathbf{x}_u$ and its output OCS. The relation between δ and δ' is explained in Equation 6.7.

Clearly, the instance-wise EMC is similar to the EMC in Formula (3) while only considers one unlabeled instance $\mathbf{x}_u$ and a certain unorder class set A_0. If both T_0' and T_0'' are "good" approximations over all the OCS $S_0 \sim A_0$, then the instance-wise EMC $\kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0')$ and $\kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0'')$ on both sub-samples should be approximately equal on each unlabeled instance $\mathbf{x}_u$. In other words, the quantity $\kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0') - \kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0'') \approx 0$ for all $u \notin L$, which can

be validated using a t-test with a hypothesis that the mean of this quantity is 0. If the t-test does not reject the hypothesis, we may consider both T_0' and T_0'' as "good" approximations over all the OCS $S_0 \sim A_0$, and take $A_0' \cup A_0''$ as the sub-sample over all the OCS $S_0 \sim A_0$ and only considers the OCS $S_0 \in T_0' \cup T_0''$. The conditional probability $P(S_0|A_0, \mathbf{x}_0)$ of the OCS $S_0 \in T_0' \cup T_0''$ should also be normalized to exclude the OCS not in the sub-sample $T_0' \cup T_0''$; otherwise, we increase m and repeat the scheme until the t-test does not reject the hypothesis.

6.2.2.4 Approximating the OCS change of an instance The third important quantity to be estimated is the OCS related model change $\delta(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}$. We calculate the model change by observing and assessing changes in the ordered class sets (OCSs) assigned for each unlabeled example $\mathbf{x}_u, i$ by models $\mathbf{f}_L$ and $\mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}$. More formally, we express the model change as:

$$\delta(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}) = \sum_{\mathbf{x}_u} \delta'(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle,\mathbf{x}_u}) \tag{6.6}$$

where δ' reflects the OCS change observed on a specific unlabeled example $\mathbf{x}_u$ and its output OCS.

The OCS change can be easily measured as the absolute ranking change on all k classes of $\mathbf{x}_u$. Formally, we define a function $\text{rank}(\mathbf{f}, \mathbf{x}, c)$ which returns the ranking of class c in the output $\mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\}$. Therefore, the OCS change $\delta'(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}, \mathbf{x}_u)$ can be calculated as $\sum_{i=1}^{k} ||\text{rank}(\mathbf{f}_L, \mathbf{x}_u, i) - \text{rank}(\mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}, \mathbf{x}_u, i)||$. However, such estimation is not perfect: it takes all the k classes equally. This is inconsistent with the fact: the changes of classes on higher rankings should be emphasized. For example, if the class on the first ranking changes, the predicted class label will also change. To address this problem, we introduce Discounted Cumulative Gain (DCG) [Järvelin and Kekäläinen, 2002]. Briefly, DCG discounts the change of a class over a $\log$ expression of its ranking, which understates the changes of classes on lower rankings. Formally, the OCS change $\delta'(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}, \mathbf{x}_u)$ with DCG can be calculated as:

$$\delta'(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}, \mathbf{x}_u) = \sum_{i=1}^{k} \frac{||\text{rank}(\mathbf{f}_L, \mathbf{x}_u, i) - \text{rank}(\mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}, \mathbf{x}_u, i)||}{\log_2[1 + \text{rank}(\mathbf{f}_L, \mathbf{x}_u, i)]} \tag{6.7}$$

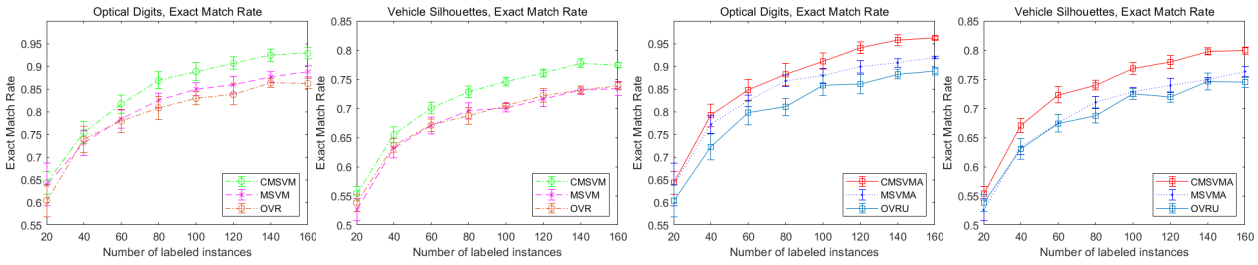

Figure 15: Performance (EMR) regarding different labeled instance numbers on two synthetic datasets.

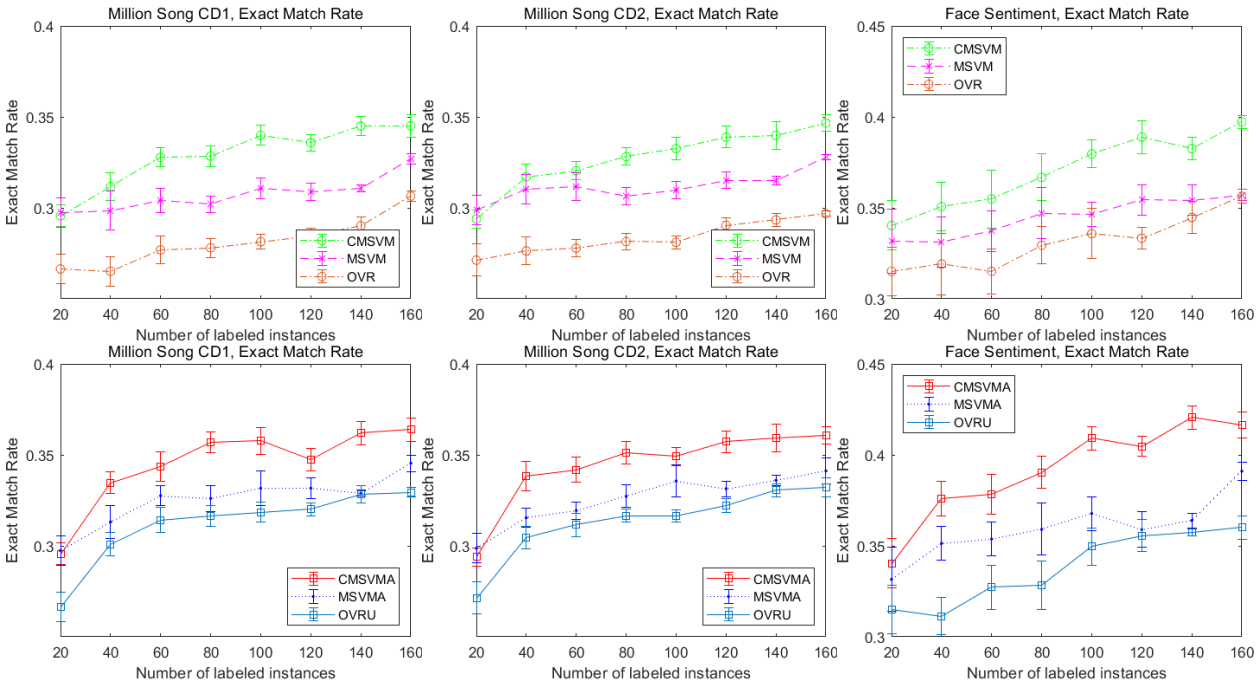

Figure 16: Performance (EMR) regarding different labeled instance numbers on three real-world datasets.

6.3 Experiments and results

We test our approach on synthetic and real-world data. The two synthetic datasets are adapted from two UCI multi-class classification datasets where the OCS are simulated; the three real-world

Dataset Name	# of Instances	# of Features	# of Classes	Size of OCS
Vehicle Silhouettes	946	18	4	Simulated
Optical Digits	5620	64	10	Simulated
Million Song CD1	35409	90	13	1∼2
Million Song CD2	89073	90	15	1∼2
Face Sentiment	584	256	4	1∼4

Table 3: Properties of all datasets (three synthetic and three real-world) in experiments.

datasets contain OCS that are assessed by human annotators and are extracted directly.

6.3.1 Experimental settings

For two synthetic datasets adapted from UCI Vehicle Silhouettes and Optical Digits datasets, we take $\frac{1}{3}$ of data instances to train an AMSVM with class labels only. For each instance on the remaining $\frac{2}{3}$ of the dataset, we can obtain the projections of all classes from the AMSVM. Then we can obtain the probabilities of all classes by applying a soft-max function to the projections. All the classes whose probability is greater than 0.05 are included in the OCS of this instance. In the following experiments we use only the remaining $\frac{2}{3}$ of instances, and retain the feature vector and OCS of each instance for training and testing.

For three real-world datasets, we use two Million Song datasets (CD1 and CD2) [Bertin-Mahieux et al., 2011] and one Face Sentiment data [Mozafari et al., 2012]. Each Million Song dataset is a collection of songs. In each dataset, the feature vector of each instance contains the timbre information of this song, the OCS of each instance contains one or two classes indicating the genre that this song likely belongs to. Please notice that each song can only belong to one genre, and the OCS of this song just indicates the competing choices of genres. In Face Sentiment data, the feature of each instance is a 128×120 gray-scale image of a facial expression, where we extract 256 features using a convolutional neural network. The class label of each instance indicates the sentiment of facial expression. However, such class labeled is annotated by 9 human annotators.

Therefore, we may sort the classes according to their vote numbers in the descending order, and take such ordered set of classes as the OCS for each instance. The basic properties of two synthetic datasets and three real-world datasets are summarized in Table 3.

To demonstrate the benefits of our multi-class classifier incorporated with ordered class set (OCS) and our expected model change (EMC) active learning strategy, we compare it with some existing multi-classifiers with and without an active learning strategy, including: (1) one-vs-rest classifier trained only on class labels, (2) one-vs-rest classifier trained only on class labels with uncertainty sampling active learning strategy, (3) approximate multi-class SVM (AMSVM) trained only on class labels, (4) AMSVM trained only on class labels with EMC active learning strategy (EMC can also be applied to multi-class classifier with class labels only by taking the class label as an OCS of size 1), and (5) our multi-class classifier incorporated with ordered class set (OCS), yet without active learning. The details of all methods in the experiments are as follows:

OVR: k one-vs-rest binary classifiers trained *independently*, one for each class; The instances to be labeled next are selected randomly; (k is the number of classes in the dataset, sic passim.)

OVRU: k one-vs-rest binary classifiers trained *independently*, one for each class; The instances to be labeled next are selected using least confident uncertainty sampling [Settles et al., 2008a] active learning strategy;

MSVM: Approximate multi-class SVM (AMSVM) where k one-vs-rest binary classifiers are trained *jointly*, one for each class; The instances to be labeled next are selected randomly;

MSVMA: Approximate multi-class SVM (AMSVM) where k one-vs-rest binary classifiers are trained *jointly*, one for each class; The instances to be labeled next are selected using our EMC active learning strategy;

CMSVM: Our multi-class classifier incorporated with OCS where k one-vs-rest binary classifiers are trained *jointly*, one for each class; The instances to be labeled next are selected randomly;

CMSVMA: Our multi-class classifier incorporated with OCS where k one-vs-rest binary classifiers are trained *jointly*, one for each class; The instances to be labeled next are selected using our EMC active learning strategy.

All data sets before the learning are split into the training and test set (using $\frac{2}{3}$ and $\frac{1}{3}$ of all data instances). We evaluate the performance of the different methods by calculating the exact match

rates (EMR) of all the classifiers achieve on the test data. Exact match rate calculates the ratio of data instances, whose prediction is identical to its class label, over all data instances. The learning considers the training data only, and the EMR is always calculated on the test set. We also repeat the splitting and learning steps 24 times. The average EMR (Y-axis) of different classifiers on two synthetic datasets and three real-world datasets regarding increasing sizes (X-axis) of the training sets is reported in Figure 15 and Figure 16 respectively.

6.3.2 Experimental results

Figure 15 and Figure 16 shows the benefit of our multi-class classifier incorporated with OCS and our EMC active learning strategy on two synthetic datasets and three real-world datasets both individually and jointly:

On all the five datasets, *CMSVM* outperforms *MSVM* and *OVR*; *CMSVMA* outperforms *MSVMA* and *OVRU*. These two groups of comparisons show that our multi-class classifier incorporated with OCS can improve the performance compared with models using only class label information at the same training size.

Also, on all the five datasets, *CMSVMA* outperforms *CMSVM*; *MSVMA* outperforms *MSVM*. These two groups of comparisons show that our EMC active learning strategy can improve the performance compared with models using only random sampling at the same training size.

Overall, on all the five datasets, the model *CMSVMA*, which is the combination of our multi-class classifier incorporated with OCS and our EMC active learning strategy, achived the highest performance. This validate the effectiveness of our multi-class classifier incorporated with OCS and our EMC active learning strategy jointly.

6.4 Summary

Ordered class set (OCS) is a special enriched label-related feedback arising in multi-class classification settings that can be easily obtained from human annotators at an insignificant cost and can help to reduce the annotation efforts. In this chapter, we proposed a new framework

for learning multi-class classification models from human feedback that utilizes OCS and a novel compatible active learning strategy: expected model change (EMC). Our results show that our learning framework (1) is able to learn more efficiently and from a smaller number of labeled instances than existing methods (2) is better than models that rely on OCS or active learning individually.

7.0 Active Learning of Multi-label Ranking, and Multi-label Classification Models with Permutation Subsets

7.1 Introduction

The work covered in this chapter is currently under review. Multi-label classification models are typically learned from annotated data in which every data instance is associated with one label vector, where each scalar is a binary value indicating whether the label is relevant to the instance. In this chapter, we explore a new form of enriched label-related feedback for multi-label classification problems: permutation subsets. Instead of a label vector, each data instance is associated with a totally ordered subset over all the labels, indicating the total orderings of the relevant labels of this instance according to their confidences. The labels not in the permutation subset are considered irrelevant to the instance. For example, in an animal image labeling task, where each image depicts an animal. The annotator can certain that this animal is orange and not dotted, while s/he just suspects the animal striped since the image is fuzzy. Since the learning of multi-label classification models with permutation subsets is identical to the learning of multi-label ranking models, we propose in this chapter a two-stage active learning framework for multi-label ranking the can be combined with existing multi-label classification models.

Multi-label ranking models, where the model assigns an ordered set of labels to data instances can be designed and learned in many different ways. A typical multi-label ranking model projects all possible labels one may assign to an instance into a real-valued space that reflects their rankings. However, such a model often assumes individual label projections are independent, hence, it ignores the dependencies that may exist among labels. In this work, we explore an alternative multi-label ranking model that relies on (1) a multi-label classification model that first selects an unordered set of labels for a data instance, and, (2) a label ranking model that orders the selected labels post-hoc. One advantage of such model is that it can use a variety of existing multi-label classification models in its first step. Another advantage, is that the label ranking model (used in the second stage), orders only labels chosen by the first model, hence it can properly reflect various label dependencies incorporated into the first model.

To translate the above idea into a working framework, we develop a new max-margin multi-label ranker to order post-hoc the output of an existing multi-label classifier. As data instances may not be initially labeled, we also explore active learning strategies to reduce the effort to annotate such data. In this chapter, we develop a new active learning strategy that considers the relevance and ordering (ranking) of the labels by calculating the expected model change (EMC) for an unlabeled instance. The EMC estimates of the model change for the different rankings of the relevant labels one can assign to the instance. Since the calculation of the expected change requires one to enumerate all possible rankings for subsets of labels, we propose a new approximation techiniques that reduces the number of the rankings to consider, making the process more efficient. To prevent the re-training of "add-one" models when adding an unlabeled instance and a possible ranking of all labels into the labeled data, which is typically inefficient and required for traditional EMC strategy, we also train the add-one models incrementally from the current model instead of from scratch. These two techniques remarkably reduce the time consumption.

We experiment with our new active learning framework for multi-label ranking combined with existing multi-label classifiers on both synthetic and real-world datasets. We evaluate two aspects of our solution: (1) its ability to find the correct labels and (2) its ability to properly rank these labels. We show the effectiveness of such an active learning framework in reducing the annotation effort in both tasks by comparing them with (1) our auxiliary max-margin multi-label ranker combined with existing multi-label classifiers without active learning and (2) existing multi-label rankers.

7.2 Methodology

In this section, we start by first defining the problem of learning of multi-label ranking model and propose a simple two-stage model for the problem. The model consists of a multi-label classifier and label ranker models and their composition. Since there are many different multi-label classification we focus on and present an multi-label ranker model responsible for ordering the labels selected by the existing multi-label classifier. After that we develop an active learning framework that builds a multi-label ranker by actively querying an annotator who provides

feedback to the framework by assessing instances with a permutation subset (rankings of relevant labels). We show how the two-step model that consists of a mutli-label classifier and ranker can be embedded in the active learning framework to improve the model by wisely selecting the examples to be assessed next. The criterion used to choose from among unlabeled candidate instances is based on the highest expected change in relevant labels and their rankings. Since the calculation of the expected change in the labels and rankings is nontrivial, we present our solutions to the following problems: (1) how to model the distribution over all the possible permutation subsets for calculating the expected change, and (2) how to speed up training via sub-sampling technique when adding one unlabeled instance and a possible permutation subset.

7.2.1 Problem

Our objective is to learn a multi-label ranking model $\mathbf{f} : X \rightarrow \mathbf{S}$, where $X \in \mathbb{R}^d$ is the input space and $\mathbf{S}$ represents the space of the permutation label subsets. The permutation subset $S^{(i)}$ reflects the rankings of the relevant labels in terms of their importance to the instance among all the K labels. The permutation subset $S^{(i)}$ is formed by a non-empty subset of K labels indicating the descending ordering of the relevant labels. The labels not in the permutation subset are considered irrelevant to the instance by the annotator. For example, in a 4-label setting, a permutation subset $\langle 3, 2 \rangle$ indicates the 3rd label is the most relevant to the instance, the 2nd label is the second most relevant, and the other two labels are irrelevant.

7.2.2 The model

The model of $\mathbf{f}$ that assigns a set of ordered labels to instances can be built in many different ways. In this work we adopt a two-step process covered with two different models to define it: $\mathbf{f} = \langle \mathbf{g}, \mathbf{h} \rangle$. The first model is a multi-label classifier $\mathbf{g} : X \rightarrow Y$ where $Y = \{0, 1\}^K$ is the space of the label vector. Such a classifier determines whether a specific label $y_j^{(i)}$ in the label vector $\mathbf{y}^{(i)}$ is relevant to the instance $\mathbf{x}^{(i)}$ or not ($y_j^{(i)} = 1$ indicates relevant). The second model is an multi-label ranker $\mathbf{h} : Y \rightarrow \mathbf{S}$ that determines the ordering of the relevant labels in $\mathbf{y}^{(i)}$ and outputs it as $S^{(i)}$. A brief illustration of this multi-label ranking model $\mathbf{f}$ is in **Figure 17**.

We note that a large body of research work in recent years has focused on the multi-label

classification problem, and many different multi-label classification models have been proposed and developed. Hence our goal in this work is not to invent a new multi-label classification model, but to utilize the existing models in our two-step multi-label ranking model.

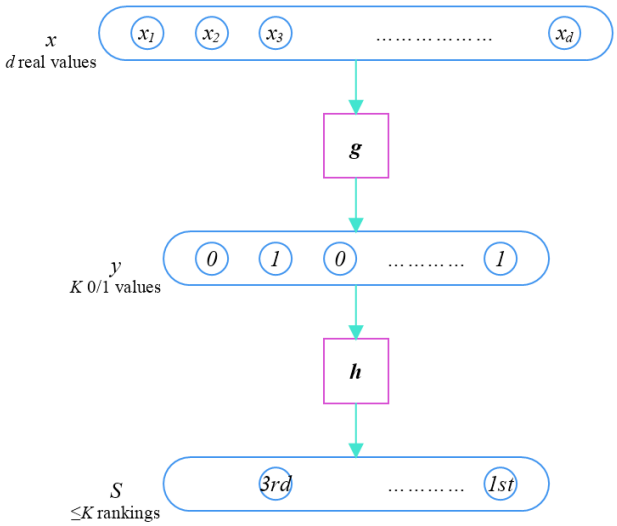

Figure 17: A two-stage multi-label ranking model **f** consisting of a multi-label classifier **g** and an auxiliary multi-label ranker **h**.

7.2.3 An auxiliary max-margin multi-label ranker

Suppose we already have access to a multi-label classifier **g** that outputs a label vector $\mathbf{y}^{(i)}$ which determines whether a label is relevant to the instance (in the permutation subset) or not. Then we can train an auxiliary max-margin multi-label ranker **h** on the label vectors such that, for each instance, the projection of a label in the permutation subset should be higher than all other labels that rank lower in the permutation subset. More formally, suppose that we have already obtained a label vector $\mathbf{y}^{(i)}$ where $y_j^{(i)} = 1$ indicates label j is included in the permutation subset. Now, we aim to obtain K different projection mappings $h_1, h_2, \ldots, h_K$, one for each label, that reflect their order in the permutation subset $S^{(i)}$. We can encode this aim by trying to enforce the following constraints: $h_j(\mathbf{y}^{(i)}) > h_l(\mathbf{y}^{(i)}) \Leftrightarrow r(S^{(i)}, j) < r(S^{(i)}, l)$, that is, the projection h_j of label j should be higher than the projection h_l of any label l such that the ranking $r(S^{(i)}, j)$ of label j in $S^{(i)}$ is beyond the ranking $r(S^{(i)}, l)$ of label l. Particularly, if $j \notin S^{(i)}$, $r(S^{(i)}, j) = |S^{(i)}| +$

1. Therefore, our auxiliary max-margin multi-label ranker can be formulated as the following optimization problem:

$$\min_{W,\Xi} \sum_{j=1}^{K} R(\mathbf{w}_j) + C \sum_{i=1}^{N} \sum_{j=1}^{K-1} \sum_{l=j+1}^{K} \xi_{jl}^{(i)}$$
$$z_{jl}^{(i)} (\mathbf{w}_j - \mathbf{w}_l)^\top \phi(\mathbf{y}^{(i)}) \geq 1 - \xi_{jl}^{(i)}$$
$$\xi_{jl}^{(i)} \geq 0$$

where $\mathbf{w}_j \subset W$ is the model parameter of h_j; $R(\mathbf{w}_j)$ is the regularization term of h_j; $\mathbf{y}^{(i)}$ is the label vector of instance i obtained from the given multi-label classifier $\mathbf{g}$; $\phi(\cdot)$ is the projection of kernel space; $z^{(i)}_{j,l}$ is the ternary value indicating the comparison of the rankings between label j and l: 1 if $r(S^{(i)}, j) < r(S^{(i)}, l)$, and -1 if $r(S^{(i)}, j) > r(S^{(i)}, l)$, and 0 otherwise; $\xi^{(i)}_{j,l} \in \Xi$ is the slack variable penalizing when the comparison between $h_j(\mathbf{y}^{(i)})$ and $h_l(\mathbf{y}^{(i)})$ violates their rankings in $S^{(i)}$.

7.2.4 Active learning for multi-label ranking framework

The challenge we want to address next is how to actively learn our multi-label ranking model $\mathbf{f}$. The core of any active learning strategy is a schema to select examples to be queried next. In this work, we propose and experiment with a strategy called expected model change [Tong and Koller, 2000, Settles et al., 2008b] that measures the potential of an unlabeled data instance to change the model by estimating its impact on predictions. In this section, we first show how the expected model change of an unlabeled instance can be calculated by considering all the possible permutation subsets of a label vector and all the possible label vectors of this instance. After that, we tackle three related problems: (1) how to obtain the probability of a label vector given an unlabeled instance (2) how to obtain the probability of a specific permutation subset given ; (3) how to measure the change of the model given an unlabeled instance and one of its permutation subset.

7.2.4.1 Expected model change (EMC) Our objective is to assess how much impact the annotation of a currently unlabeled example $\mathbf{x}^{(0)}$ can make if with a permutation subset. Let $\Delta(L, \mathbf{x}^{(0)})$ be a measure of this impact. In this work, we assess the impact in terms of the expected model change, and an unlabeled instance with the highest expected model change is selected for the labeling first. Formally, we define the expected model change of an unlabeled instance $\mathbf{x}^{(0)}$ as:

$$\Delta(L, \mathbf{x}^{(0)}) = \sum_{\mathbf{y}^{(0)}} P(\mathbf{y}^{(0)}|\mathbf{x}^{(0)})\delta_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle) + t \sum_{\mathbf{y}^{(0)}} P(\mathbf{y}^{(0)}|\mathbf{x}^{(0)})[\sum_{S^{(0)} \in \mathbf{S}} P(S^{(0)}|\mathbf{y}^{(0)})\delta_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle)] \quad (7.1)$$

where $\delta_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle)$ denotes the classification change induced by assigning a label vector $\mathbf{y}^{(0)} \in \{0, 1\}^K$ to the unlabeled example $\mathbf{x}^{(0)}$ on the multi-label classifier $\mathbf{g}$; $\delta_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle)$ denotes the ranking change induced by adding a permutation subset $S^{(0)}$ and its corresponding label vector $\mathbf{y}^{(0)}$ into the multi-label ranker $\mathbf{h}$; t is the coefficient that balances two kinds of model changes. In order to calculate the expected model change the following quantities defining it need to be estimated: (1) the conditional probabilistic distribution $P(\mathbf{y}^{(0)}|\mathbf{x}^{(0)})$ of the label vector $\mathbf{y}^{(0)}$ given an unlabeled instance $\mathbf{x}^{(0)}$, (2) the conditional probabilistic distribution $P(S^{(0)}|\mathbf{y}^{(0)})$ of the permutation subset $S^{(0)}$ given a label vector $\mathbf{y}^{(0)}$, (3) the classification change $\delta_g(L, L \cup \langle \mathbf{x}^{(0)}$ and (4) the ranking change $\delta_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle)$. To prevent enumerating all the permutation subsets when calculating the permutation subset distribution, we propose an approximation subsampling over all the permutation subsets. To prevent the re-training of "add-one" models when adding an unlabeled instance, a possible label vector and a possible permutation subset into the labeled data, which is typically inefficient and required for traditional EMC strategy, we train the add-one models incrementally from the current model instead of from scratch. These two techniques remarkably reduce the time consumption. We present the details of our solutions to these next.

7.2.4.2 Finding the MLE of the label vector The first quantity to be estimated is the conditional probabilistic distribution $P(\mathbf{y}^{(0)}|\mathbf{x}^{(0)})$ over the label vector $\mathbf{y}^{(0)}$ given an unlabeled instance $\mathbf{x}^{(0)}$. To simplify the calculation, we may consider only the maximum likelihood estimate (MLE) $\hat{\mathbf{y}}^{(0)}$ and ignores all the other label vectors instead of finding the conditional distribution

over $\mathbf{y}^{(0)}$. The MLE $\hat{\mathbf{y}}^{(0)} = \mathbf{g}(\mathbf{x}^{(0)})$ can be directly obtained from the existing multi-label classifier $\mathbf{g}$. After that, we let $P(\mathbf{y}^{(0)} = \hat{\mathbf{y}}^{(0)}|\mathbf{x}^{(0)}) = 1$ and $P(\mathbf{y}^{(0)} \neq \hat{\mathbf{y}}^{(0)}|\mathbf{x}^{(0)}) = 0$.

7.2.4.3 Estimating the conditional probability of a permutation subset The second quantity to be estimated is the conditional probability $P(S^{(0)}|\mathbf{y}^{(0)})$ of each permutation subset $S^{(0)}$ conforming to the label vector $\mathbf{y}^{(0)}$. Since it is hard for us to directly estimate $P(S^{(0)}|\mathbf{y}^{(0)})$ due to the correlations among labels, we propose an approximation using $\mathbf{y}^{(0)}$ of the multi-label classifier and the projections $\mathbf{h}(\mathbf{y}^{(0)})$ of the auxiliary multi-label ranker via Discounted Cumulative Gain (DCG) [Järvelin and Kekäläinen, 2002]. Briefly, DCG discounts the weight of a label over a $\log$ expression of its ranking, which understates the weight of labels on lower rankings. The weight of each label is determined by the projections $\mathbf{h}(\mathbf{y}^{(0)})$ of the multi-label ranker. Since we only need to consider the labels in $S^{(0)}$, the labels whose corresponding scalar in $\mathbf{y}^{(0)}$ is 0 are not in $S^{(0)}$ and will not be considered. Formally, the probability $P(S^{(0)}|\mathbf{x}^{(0)})$ can be approximated as:

$$P(S^{(0)}|\mathbf{y}^{(0)}) \propto \exp(\sum_{j=1}^{K} \frac{y_j^{(0)} f_j(\mathbf{y}^{(0)})}{\log_2(1 + r(S^{(0)}, j))})$$

Again, $r(S^{(0)}, j)$ returns the ranking of label j in $S^{(0)}$. If $j \notin S^{(0)}$, $r(S^{(0)}, j) = |S^{(0)}| + 1$.

It appears the conditional probability $P(S^{(0)}|\mathbf{y}^{(0)})$ is well approximated. However, there is an inevitable fact: each $S^{(0)}$ is a permutation of all the labels j such that $y_j^{(0)} = 1$. This indicates that, given a label vector $\mathbf{y}^{(0)}$, the number of permutation subsets $S^{(0)}$ such that $S^{(0)} \sim \mathbf{y}^{(0)}$, in other words $S^{(0)}$ conforms to $\mathbf{y}^{(0)}$, is actually $||\mathbf{y}^{(0)}||_1 = \sum_{j=1}^{K} y_j^{(0)}$. When calculating the classification change $\delta_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle)$ and the ranking change $\delta_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle)$, we also need to calculate all the instance-wise classification changes $\delta_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle)$ and instance-wise ranking changes $\delta_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle)$ by enumerating all the unlabeled instances u as follows:

$$\begin{aligned} \delta_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle) &= \sum_{u \notin L} \delta'_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle, u) \\ \delta_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle) &= \sum_{u \notin L} \delta'_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle, u) \end{aligned} \tag{7.2}$$

where δ'_g and δ'_h reflects the classification and ranking change observed on a specific unlabeled example u respectively. Therefore, the complexity of our EMC strategy for a given unlabeled instance $\mathbf{x}^{(0)}$ is $O(U||\mathbf{y}^{(0)}||_1!) \leq O(UK!)$, where U is the number of unlabeled instances. Clearly,

it is intractable to calculate the conditional probability $P(S^{(0)}|\mathbf{y}^{(0)})$ for all the $S^{(0)}$. To reduce the number of permutation subsets to enumerate, a straightforward method is to do random sub-sampling over all the $S^{(0)}$ conforming to $\mathbf{y}^{(0)}$. However, such method introduces another problem: is such a sub-sample a "good" approximation over all the $S^{(0)}$? That is, is the EMC obtained using this sub-sample similar to the EMC obtained by considering all the $S^{(0)}$ such that $S^{(0)} \sim \mathbf{y}^{(0)}$? To solve this problem, we propose the following sub-sampling scheme:

(I) We create two random sub-samples $T_1^{(0)}$ and $T_2^{(0)}$ over all the $S^{(0)}$ such that: (1) $S^{(0)} \in T_1^{(0)} \Rightarrow S^{(0)} \sim \mathbf{y}^{(0)}$ and $S^{(0)} \in T_2^{(0)} \Rightarrow S^{(0)} \sim \mathbf{y}^{(0)}$. In other words, both $T_1^{(0)}$ and $T_2^{(0)}$ only contains the permutation subset whose corresponding label vector is $\mathbf{y}^{(0)}$; (2) $T_1^{(0)} \cap T_2^{(0)} = \emptyset$, and (3) $|T_1^{(0)}| = |T_2^{(0)}| = m$ where m is a small number.

(II) We define an instance-wise EMC $\kappa(L, \mathbf{x}^{(0)}, \mathbf{x}^{(u)}, T_1^{(0)})$ of the unlabeled instance $\mathbf{x}^{(0)}$ on an arbitrary unlabeled instance $\mathbf{x}^{(u)}$ and a sub-sample $T_1^{(0)}$ as follows:

$$\kappa(L, \mathbf{x}^{(0)}, u, T_1^{(0)}) = \sum_{\mathbf{y}^{(0)}} P(\mathbf{y}^{(0)}|\mathbf{x}^{(0)}) \delta'_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle, \mathbf{x}^{(u)}) + t \sum_{\mathbf{y}^{(0)}} P(\mathbf{y}^{(0)}|\mathbf{x}^{(0)}) [\frac{1}{Z} \sum_{S^{(0)} \in T_1^{(0)}} P(S^{(0)}|\mathbf{y}^{(0)}) \delta'(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle, \mathbf{x}^{(u)})]$$

where $u \notin L$; $T_1^{(0)}$ is a sub-sample only contains the permutation subsets $S^{(0)} \sim \mathbf{y}^{(0)}$. $Z = \sum_{S^{(0)} \in T_1^{(0)}} P(S^{(0)}|\mathbf{y}^{(0)})$ is the partition function. δ'_g and δ'_h reflects the classification and ranking change observed on a specific unlabeled example u respectively. Similar to **Section 7.2.4.2.2**, we only consider the maximum likelihood estimate (MLE) $\hat{\mathbf{y}}^{(0)}$ and ignores all the other label vectors instead of finding the conditional distribution over $\mathbf{y}^{(0)}$ to simplify the calculation.

Clearly, the instance-wise EMC is similar to the EMC in **Formula 7.1** while considering only one unlabeled instance u and a certain sub-sample $T_1^{(0)}$ over all the $S^{(0)}$.

(III) If both $T_1^{(0)}$ and $T_2^{(0)}$ are "good" approximations for all the $S^{(0)}$, then the instance-wise EMC $\kappa(L, \mathbf{x}^{(0)}, u, T_1^{(0)})$ and $\kappa(L, \mathbf{x}^{(0)}, u, T_2^{(0)})$ on both sub-samples should be approximately equal for any unlabeled instance u. In other words, the assertion $\kappa(L, \mathbf{x}^{(0)}, u, T_1^{(0)}) - \kappa(L, \mathbf{x}^{(0)}, u, T_2^{(0)}) \approx 0$ should hold for all $u \notin L$, which can be validated using a t-test with a hypothesis that the mean of such quantity is 0. If the t-test does not reject the hypothesis, we may consider both $T_1^{(0)}$ and $T_2^{(0)}$ as "good" approximations over all the $S^{(0)}$, and take $T_1^{(0)} \cup T_2^{(0)}$ as the sub-sample over all

the $S^{(0)}$, and only consider the permutation subsets $S^{(0)} \in T_1^{(0)} \cup T_2^{(0)}$. The conditional probability $P(S^{(0)}|\mathbf{y}^{(0)})$ of all the permutation subset $S^{(0)} \in T_1^{(0)} \cup T_2^{(0)}$ should also be re-normalized to exclude the permutation subsets not in the sub-sample $T_1^{(0)} \cup T_2^{(0)}$; otherwise, we increase m and repeat from step (I) of this scheme until the t-test does not reject the hypothesis. By applying the sub-sampling technique above, the complexity of EMC for a given unlabeled data instance $\mathbf{x}^{(0)}$ is reduced to $O(Um)$.

7.2.4.4 Approximating the change on the permutation subset of an instance The third and fourth important quantities to be estimated are the classification change $\delta_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle)$ and the ranking change $\delta_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle)$. We calculate these quantities by observing and assessing changes for each unlabeled example u. More formally, we express the classification change and the ranking change the same as in **Formula 7.2**:

$$\delta_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle) = \sum_{u \notin L} \delta'_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle, u)$$
$$\delta_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle) = \sum_{u \notin L} \delta'_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle, u)$$

where δ'_g and δ'_h reflects the classification and ranking change observed on a specific unlabeled example u, respectively.

The first concern related to the calculation of model change is the time complexity of training the ranking model $\mathbf{f}_{L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)}, S^{(0)} \rangle} = \langle \mathbf{g}_{L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle}, \mathbf{h}_{L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle} \rangle$. This is a non-trivial concern since, for a given current model $\mathbf{f}_L = \langle \mathbf{g}_L, \mathbf{h}_L \rangle$, we need to train $O(Um)$ different add-one models. However, this will not be a problem if the given multi-label classifier $\mathbf{g}$ and our auxiliary multi-label ranker $\mathbf{h}$ can be trained via gradient-based algorithms: by setting the model parameter of the current model $\mathbf{f}_L$ as the initial value, the training of the add-one model can be finished incrementally, much faster than training from scratch. Clearly, our auxiliary multi-label ranker is a max-margin model supporting gradient-based algorithms. Since the total number of constraints of our auxiliary max-margin multi-label ranker is $O(NS^2)$, where N is the number of labeled instances and S is the average size of the permutation subset of each instance, the incremental training just adds $O(S^2)$ constraints of the newly added instance into the current model.

The label change δ'_g induced by the existing multi-label classifier $\mathbf{g}$ can be easily measured as the Hamming distance between the predicted label vector on the current classifier and the add-one classifier. Briefly, let $\mathbf{y}^{(u)}$ denotes the predicted label vector of an unlabeled example u on the current classifier $\mathbf{g}_L$, and $\mathbf{y}'^{(u)}$ denotes the predicted label vector of u on the add-one classifier $\mathbf{g}_L$, then the Hamming distance between $\mathbf{y}^{(u)}$ and $\mathbf{y}'^{(u)}$ is defined as:

$$\delta'_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle, u) = \sum_{j=1}^{K} ||y_j^{(u)} - y_j'^{(u)}||$$

The ranking change δ'_h induced by the auxiliary multi-label ranker $\mathbf{h}$ can be also easily measured as the absolute ranking change on all the relevant labels. Briefly, let $S^{(u)}$ denotes the predicted permutation subset of an unlabeled example u on the current ranker $\mathbf{h}_L$, and $S'^{(u)}$ denotes the predicted permutation subset of u on the add-on ranker $\mathbf{h}_{L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle}$, then such change δ'_h in the permutation subset can be calculated as $\delta'_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle, u) = \sum_{j \in S^{(u)}} ||r(S^{(u)}, j) - r(S'^{(u)}, j)||$. Particularly, any label $j \notin S'^{(u)}$ will be ignored, since the change on such label should be treated as the label change induced by the existing multi-label classifier $\mathbf{g}$. However, such estimation is not perfect: it assumes all the relevant labels contribute equally to the change. This is however, inconsistent with the fact in multi-label ranking tasks: the ranking changes of labels on higher rankings should be emphasized. To address this problem, we use Discounted Cumulative Gain (DCG) [Järvelin and Kekäläinen, 2002] to discount the change of a label over a $\log$ expression of its ranking, which understates the changes of labels on lower rankings. Formally, the ranking change δ'_g with DCG can be calculated as:

$$\delta'_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle, u) = \sum_{j \in S^{(u)}} \frac{||r(S^{(u)}, j) - r(S'^{(u)}, j)||}{\log_2[1 + r(S^{(u)}, j)]}$$

Again, any label $j \notin S'^{(u)}$ will be ignored.

The last quantity is the coefficient t balancing two kinds of expected changes. Intuitively, the label change on one label should be greater than the ranking change on one label if any, since the label change indicates the change in both the classifier $\mathbf{g}$ and the ranker $\mathbf{h}$. Therefore, we may set $t = \frac{1}{K}$ so that we always have $\delta'_g(L, L \cup \langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle, u) \geq t\delta'_h(L, L \cup \langle \mathbf{y}^{(0)}, S^{(0)} \rangle, u)$.

Dataset	Instances	Features	Labels (Sets)	Cardinality
Emotions	593	72	6 (27)	1.9
Yeast	2417	103	14 (198)	4.2
Scene	2407	294	6 (15)	1.1
MS1	35409	90	13 (156)	1.3
MS2	89073	90	15 (210)	1.3
Faces	584	256	4 (23)	1.4

Table 4: Properties of all datasets (three synthetic and three real-world) in experiments.

7.3 Experiments and results

We test our model and active learning approach on multiple synthetic and real-world datasets. The three synthetic datasets are built from UCI multi-label classification datasets where the permutation subsets are simulated; the three real-world datasets contain permutation subsets provided by human annotators.

7.3.1 Datasets

The synthetic datasets are generated from UCI multi-label classification datasets. We generate them by taking $\frac{1}{3}$ of data instances to train a multi-label ranking model with $0/1$ label vectors only. This is possible since we can still enforce that the projections of relevant labels should be higher than the projections of irrelevant labels. After training, we apply the trained multi-label ranking model to every instance in the remaining $\frac{2}{3}$ of the dataset and calculate the rankings of all its labels. By combining the label vector and the predicted rankings, we generate permutation subsets for every instance in the remaining $\frac{2}{3}$ of the dataset. In the experiments, we use only the $\frac{2}{3}$ of data that consists of the original feature vectors and the generated permutation subsets.

The real-world datasets consists of two Million Song datasets (MS1 and MS2) [Bertin-Mahieux et al., 2011] and one Face Sentiment dataset [Mozafari et al., 2012]. Each

Million Song dataset consists of a collection of songs. In each dataset, the feature vector of an instance (song) contains the timbre information of the song, and the permutation subset of each instance contains one or two labels indicating the priorities of the genres. In Face Sentiment data, the feature of each instance is a 128×120 gray-scale image of a facial expression, where we extract 256 features using a convolutional neural network. The output of each instance indicates one sentiment of facial expression out of four provided by nine human annotators. Therefore, we may sort the output sentiment according to their vote numbers in the descending order, and take such an ordered set as the permutation assigned to each instance. The basic properties of two synthetic datasets and the three real-world datasets are summarized in Table 4.

7.3.2 Settings

To demonstrate the benefits of our model and the active learning strategy based on the expected model change for multi-label ranking models, we compare the performance of the following two models:

CTBN, a combination of the conditional tree-structured Bayesian network (CTBN) [Batal et al., 2013, Hong et al., 2014, Hong et al., 2015] multi-label classifier and our multi-label ranker. The next unlabeled instance to be labeled is selected randomly;

CTBNAct, a combination of the conditional tree-structured Bayesian network (CTBN) [Batal et al., 2013, Hong et al., 2014, Hong et al., 2015] multi-label classifier and our multi-label ranker. The next unlabeled instance to be labeled is selected using our active learning strategy based on the expected model change;

CRF, a combination of the conditional random field (CRF) [Lafferty et al., 2001, Bradley and Guestrin, 2010, Naeini et al., 2015] multi-label classifier and our multi-label ranker. The next unlabeled instance to be labeled is selected randomly;

CRFAct, a combination of the conditional random field (CRF) [Lafferty et al., 2001, Bradley and Guestrin, 2010, Naeini et al., 2015] multi-label classifier and our multi-label ranker. The next unlabeled instance to be labeled is selected using our active learning strategy based on the expected model change;

MMR, the max-margin multi-label ranker [Bucak et al., 2009] that combines the constraints

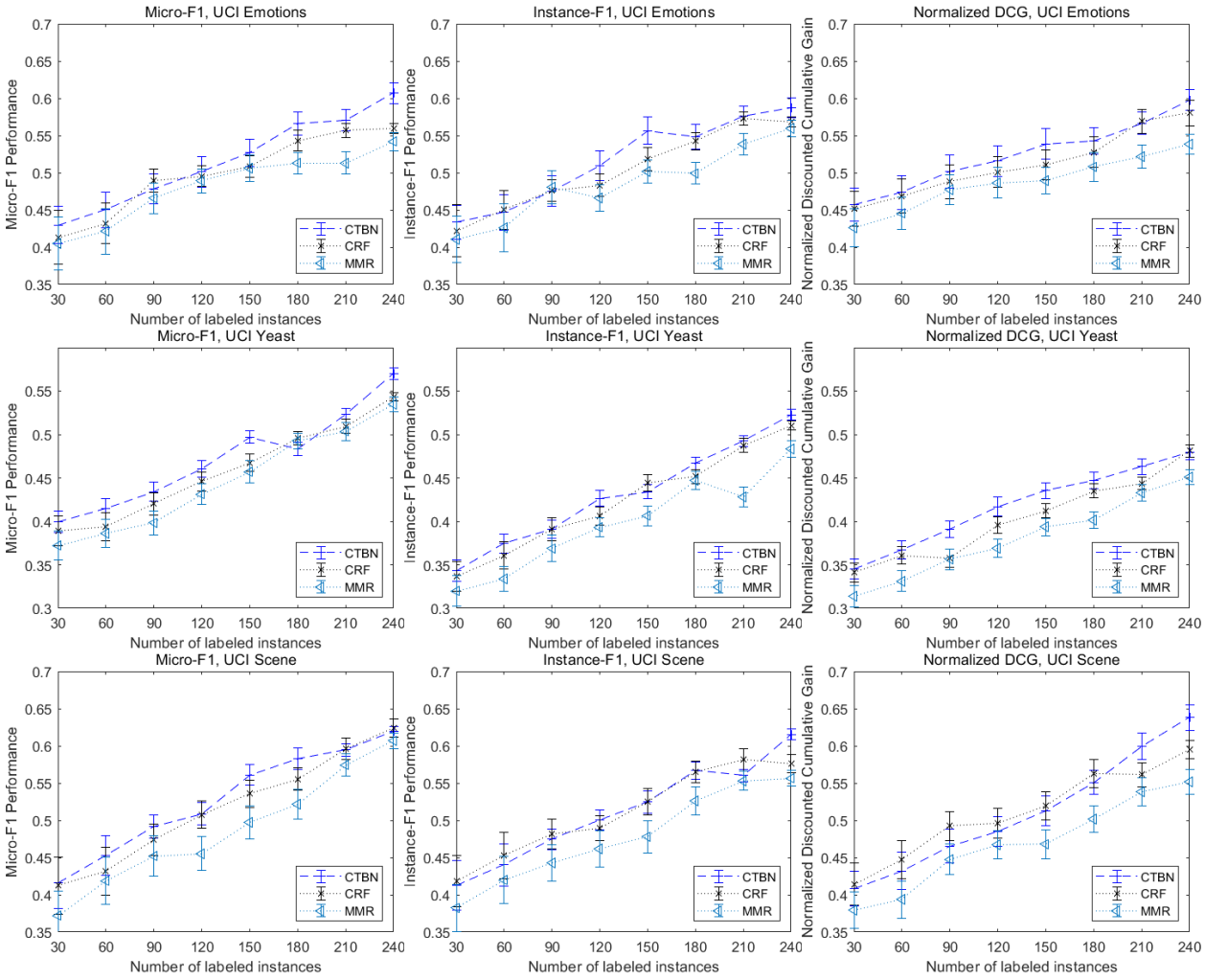

Figure 18: Performance (Micro-F1, Instance-F1, Normalized DCG) with random sampling regarding different labeled instance numbers on all synthetic datasets.

from pairwise ordering extracted from the label rankings in the permutation subset of each instance. The next unlabeled instance to be labeled is selected randomly;

MMRAct, the max-margin multi-label ranker [Bucak et al., 2009] that combines the constraints from pairwise ordering extracted from the label rankings in the permutation subset of each instance. The next unlabeled instance to be labeled is selected from our expected model change for multi-label rankers.

OBR, the online boosting multi-label ranking model [Jung and Tewari, 2018] that aggregates

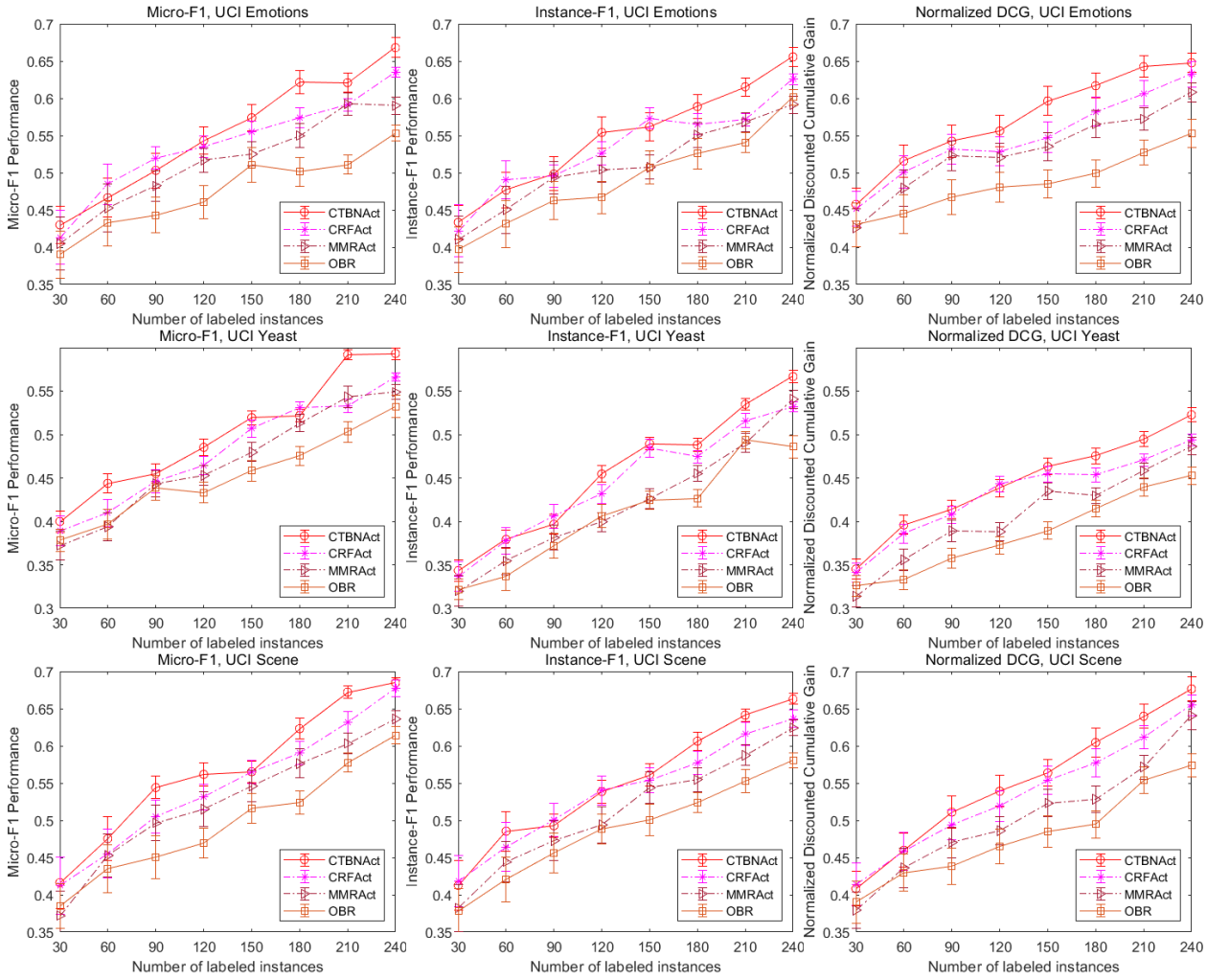

Figure 19: Performance (Micro-F1, Instance-F1, Normalized DCG) with active learning regarding different labeled instance numbers on all synthetic datasets.

the predictions of multiple weak multi-label rankers via majority votes. The next unlabeled instance to be labeled is selected based on the aggregated loss of all the weak rankers.

All data sets are split into the training and test set ($\frac{2}{3}$ and $\frac{1}{3}$ of all data instances). We evaluate the Macro-F1 (does not consider rankings), Instance-F1 (does not consider rankings), and Normalized DCG (considers rankings) on the test data regarding different numbers of labeled instances. The learning considers the training data only, and the three metrics are always calculated on the test set. We also repeat the splitting and learning steps 30 times. The average performance

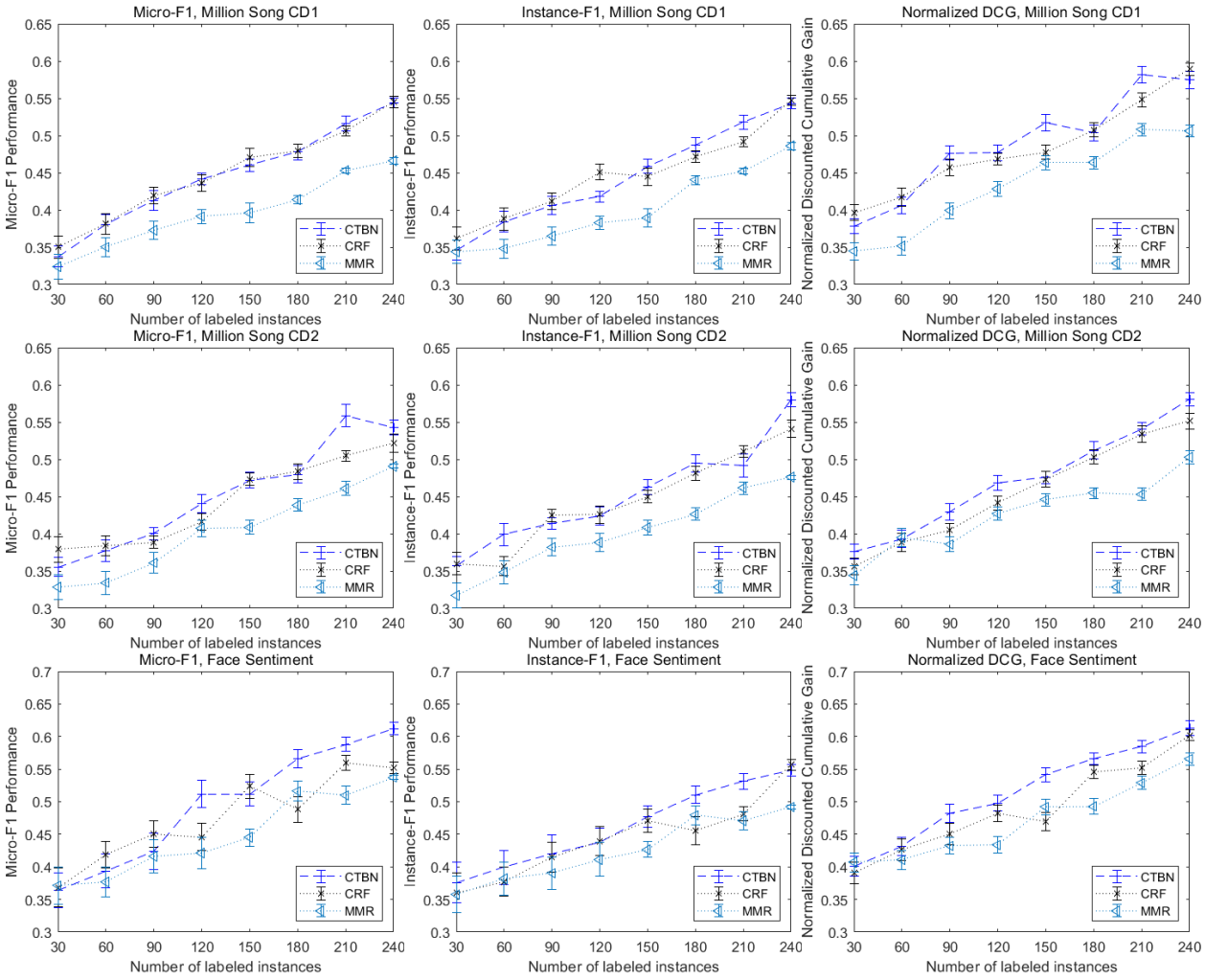

Figure 20: Performance (Micro-F1, Instance-F1, Normalized DCG) with random sampling regarding different labeled instance numbers on all real-world datasets.

(Y-axis) of different models on the datasets regarding increasing sizes (X-axis) of the training sets is reported in **Figure 18** and **Figure 20**.

7.3.3 Experimental results

Figure 18, 19, 20, and 21 show the performance of different multi-label ranking frameworks. In most experiments, *CTBN* and *CRF* outperform *MMR* and *OBR*: this shows the effectiveness

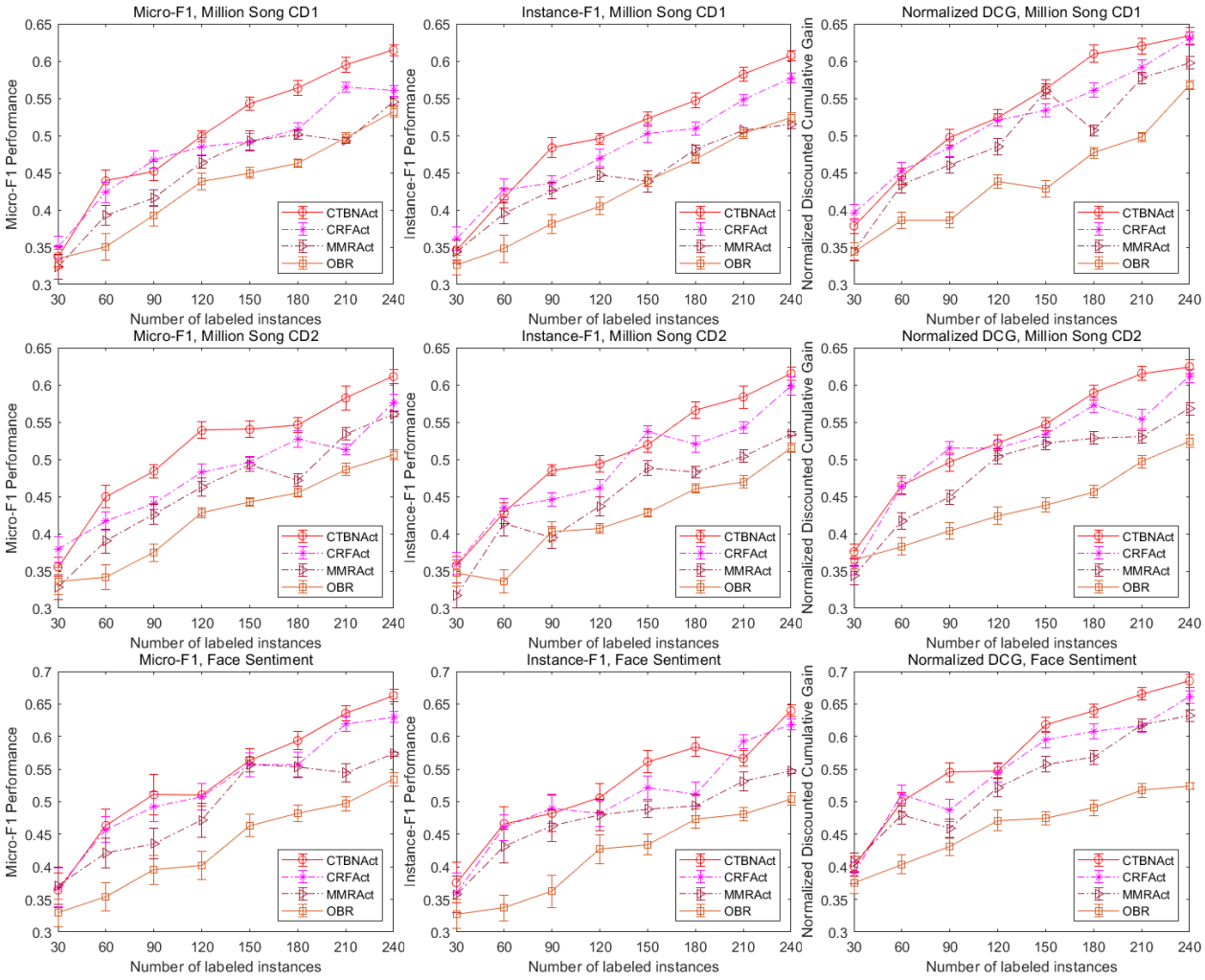

Figure 21: Performance (Micro-F1, Instance-F1, Normalized DCG) with active learning regarding different labeled instance numbers on all real-world datasets.

of the combination of the existing multi-label classifiers that model dependencies among labels and our auxiliary multi-label ranker. In all experiments, *CTBNAct* outperform *CTBN*; *CRFAct* outperform *CRF*; *MMRAct* outperform *MMR*: this shows the effectiveness of our expected model change strategy that actively selects the unlabeled instance with the highest potential to improve the classification performance. Overall, *CTBNAct*, the multi-label ranking framework combining CTBN [Batal et al., 2013, Hong et al., 2014, Hong et al., 2015], our max-margin multi-label ranker, and the expected model change strategy performs the best. This demonstrates the improved

effectiveness of the multi-label ranking model, and corresponding active learning framework.

7.4 Summary

In this chapter, we have proposed a new model for multi-label ranking that let us combine existing multi-label classifiers and our auxiliary multi-label ranker. Our multi-label ranker learns the rankings from the labels provided by a multi-label classifier. We have also proposed a new variant of expected model change (EMC) active learning strategy that actively selects the unlabeled instance with highest expected change induced by the multi-label ranking model. Our results show that our multi-label ranking model is able to utilize existing multi-label classifiers, which lets us better capture the dependencies among the labels, and learn the rankings of the selected labels more efficiently from a smaller number of labeled instances than existing multi-label rankers. Since the learning of multi-label classification models with permutation subsets is identical to the learning of multi-label ranking models, this new model can also be applied to multi-label classification tasks with permutation subsets.

8.0 Conclusions

8.1 Our contributions

In many classification tasks, training examples need to be labeled by human annotators before they can be used for learning models. The labeling process may be very costly and tedious, especially in domains where data are complex and require strong background knowledge for annotators. An example of such domains is medical diagnosis. In this thesis, we studied and proposed solutions to address the cost-sensitive learning problem, where the objective is to learn better classification models while reducing and efficiently distributing the cost of labeling. The main contributions of this work are summarized as follows:

- Overall, we proposed multiple learning methods incorporating different forms of enriched label-related feedback and complementary active learning strategy to reduce the annotation effort in binary, multi-class, and multi-label classification scenarios. In this thesis, we are mainly focused on two families of popular (maybe the most popular) enriched label-related feedback: (1) confidence of the class labels, including probabilistic scores and Likert-scale feedback, which are available in online stores, forecast of precipitation, and other tasks in our daily life; (2) orderings among the class labels, including ordered class sets and permutation subsets, which are available in differential diagnosis, object recognition, and other tasks in biomedical informatics, computer vision, and natural language processing tasks. We are also mainly focused on the variants of expected model change active learning strategy since it better estimates how an unlabeled instance will change the model if labeled. We also proposed multiple techniques to reduce the time complexity of our expected model change strategies. Our methods are general and can be applied in multiple domains, so we had experiments on simulated data and real-world data in different domains, showing the superior performance of our methods compared with existing methods.
- We presented a learning framework for binary classification problems from a form of enriched label-related feedback named probabilistic scores (or soft labels), where we ask the human annotator to provide us with, in addition to binary class label, also a probability reflecting

his/her belief in the positive class label (class 1), and incorporate this information into the learning process. Our framework was based on the following assumptions: (1) the cost of labeling is mostly in the example review. Once an example is reviewed and binary class label is given, the human annotator can give us the probabilistic score at an insignificant cost; (2) the probabilistic scores given by human may be noisy and negatively influence the learning process, therefore the methods that utilize probabilistic scores should be robust against such noise. We proposed a method derived from ordinal regression that distributes training examples to multiple discrete bins based on their values of probabilistic scores, then enforces optimization constraints on examples and bin boundaries. The main advantage of this method is the low time complexity: the number of constraints is linear to the number of training examples.

- Our method for binary classification problems from probabilistic scores requires the discretization of examples into multiple bins based on the values of probabilistic scores. Therefore, we need to determine the optimal number. Since the ordinal-regression-based method considers all the probabilistic scores in the same bin as an entity. In other words, the ordinal-regression-based method approximates the probabilistic-score distribution of each bin as a uniform distribution. That is, the ordinal-regression-based method approximate the probabilistic-score distribution in the same way as histogram. Therefore, in this thesis, we proposed utilizing the Freedman-Diaconis rule [Freedman and Diaconis, 1981] for histogram to select the optimal bin number. We showed the bin number selection can provide closely comparable performance when combined with our method for binary classification problems from probabilistic scores compared with internal cross validation.
- Our learning framework for binary classification problems from probabilistic scores can also be combined with active learning, where we select the unlabeled instance to be labeled next with the highest potential to improve the model performance, which can further reduce the annotation effort. However, we pointed out that existing active learning strategies, such as uncertainty sampling, are incompatible with probabilistic scores since probabilistic scores already contain uncertainty information. To solve this problem, we proposed an active learning strategy based on expected model change, where the unlabeled instance with the highest expected change to the predictions over the unlabeled data is selected. We also proposed

multiple approximation approaches to reduce the time complexity of our active learning strategy.

- We extended our learning framework to more complex classification scenarios: multi-class and multi-label classification. We also extended our learning framework to other forms of enriched label-related feedback: Likert-scale feedback in binary classification, probabilistic scores in multi-class classification, ordered class sets in multi-class classification, and permutation subsets in multi-label classification. For Likert-scale feedback in binary classification, we also assume that such feedback can also be obtained in negligible time compared with the review of binary labels. For probabilistic scores in multi-class classification, we assume that only the probabilistic score of the true-label class can be obtained, since obtaining the probabilistic scores of all class will take non-trivial time. For ordered class sets in multi-class classification, we assume that only the total orderings of the top few classes can be obtained. For permutation subsets in multi-label classification, we assume that only the total orderings of the relevant labels can be obtained. This is mainly because the low-ranked class or the irrelevant labels are typically of negligible probabilities, which cannot be distinguished in a short time. We also proposed the efficient and robust learning methods for all these forms of enriched label-related feedback along with compatible active learning strategies to reduce the annotation effort.
- Our method for multi-label classification problems from permutation subsets can also be applied to multi-label ranking problems. Such two-stage method can be trained efficiently and can be combined with most existing multi-label classification methods which support gradient-based training methods. We also proposed a complementary active learning strategy for multi-label ranking models inspired from expected model change, which estimates the expected change on the predicted total orderings of the relevant labels over the unlabeled data when an unlabeled instance is assumed given the total orderings of its relevant labels. This method is, to our knowledge, the first active learning strategy for multi-label ranking models.

8.2 Open questions

We have proposed approaches to learn better classification models while reducing the cost of labeling. Our approaches show superior performance compared to existing approaches. However, there are still multiple challenges and open problems that require further investigation. In the following, we briefly summarize some of these interesting new directions.

- One important problem studied in this thesis has focused on the development of active learning methods utilizing enriched label-related feedback where the agreement of the human expert with the class label has been represented by a probability of the instances belonging to that class. In parallel to our effort [Luo and Hauskrecht, 2018a, Luo and Hauskrecht, 2018b, Luo and Hauskrecht, 2019, Luo and Hauskrecht, 2020] have studied group-based active learning methods where the human expert provides feedback in terms of a proportion of positive instances for a region of the input space. An interesting open direction to investigate is a combination of the two approaches. More specifically, we can view an instance to be an infinitesimal region around that instance and soft-label probability assigned by a human to that instance to be equal to the proportion of instance replicas the human believes fall into the positive class. This view gives us the flexibility to query both the regions and individual instances the same way possibly combining the strengths of both methods. For example, one solution could be a two-stage active learning framework in which the human expert first provides the group-based feedback indicating the proportion of positive instances over larger regions of the input space. This feedback can help to find rough values of the model parameters. After that, the framework switches to specific instances rather than regions that can help to better fine-tune the values of the model parameters.
- All work in this thesis for utilizing enriched label-related feedback assumed the class and enriched label assessments were provided by one human expert. However, because of the time it may take the human to review and annotate an example, it is hard to expect one human expert to label all the instances in the dataset. To address this, instead of asking one human expert, we can ask multiple human experts for the labels. The main challenge is that different experts may have different opinions, knowledge, or biases, leading to disagreements in labels. To solve this problem, [Valizadegan et al., 2012, Valizadegan et al., 2013] proposed

a multi-expert learning framework for binary classification tasks with class labels, which modeled the model-consistency among the experts with a set of expert-specific parameters. Therefore, an interesting open research direction is to incorporate enriched label-related feedback as probabilistic scores into the multi-expert learning framework. In other words, we hope to utilize enriched label-related as probabilistic scores provided by multiple human experts, where different expert labels different part of the dataset. In this new setting, the main challenge is still model-consistency. In binary classification tasks with merely class labels, different human experts may have different biases when providing binary labels: some tend to provide more positive labels, while others tend to provide more negative labels. In binary classification tasks with probabilistic scores, the model-consistency problem becomes more complicated, since human experts may have different biases on the distribution of probabilistic scores: some may tend to provide higher scores, some may tend to provide lower scores, some may tend to provide medium values, and some may tend to provide extreme values. Another challenge of this learning framework is the noise hidden in the probabilistic scores arisen from subjective human reviews since it is well documented that humans are unable to provide consistent and precise probabilistic assessments. In other words, the multi-expert learning framework with probabilistic scores must (1) eliminate the inconsistency on the distribution of probabilistic scores among the human experts, and (2) be robust against the noise in the inaccurate probabilistic scores arisen while still utilizes the useful information in such refined feedback. Also, since probabilistic scores provide more refined information than merely class labels, the optimization of this learning framework will be more time consuming, and we have to be careful with the time complexity of the learning framework.

- Throughout the thesis, when analyzing the benefits of various human feedback strategies, we assumed the cost of labeling all instances is fixed and constant. However, we note that this assumption may not hold in practice. Briefly, data objects may be presented to the human annotator in many different formats, and some of these may be more or less human friendly. For example, if images were presented to the user as numerical matrices, these would be very hard, if not impossible, for humans to analyze and assess. Another aspect of this problem is that high dimensional objects may be harder and more time consuming to review for humans. For example, if we describe a data object (instance) using a set of attributes and

their values, it clearly takes more time to review objects with 100 attributes than objects with ten attributes. Hence an interesting open question in this context is how to properly account for the complexity of the instance and its query and how to properly model its review cost. A closely related issue to the review cost model is how to re-represent or transform data instances to minimize their review cost. Assuming a low dimensional data instance is easier (less costly) to review and annotate than a high dimensional data instance, one possible approach to reduce the annotation cost could use feature selection methods [Guyon and Elisseeff, 2003, Hauskrecht et al., 2005, Hauskrecht et al., 2007] that would automatically restrict the features to present to the user to the most important ones.

- As noted above, the representation of data objects for review purposes can make a big difference in their review cost. However, the same (human-friendly) representation may not be optimal for building machine learning models. Take for example, complex time-series data from Electronic Health Records (EHRs) with thousands of clinical variables and related event prediction tasks [Hauskrecht et al., 2013, Hauskrecht et al., 2016, Liu and Hauskrecht, 2019]. In general, one can apply many different ways to process and featurize the time series to support prediction tasks, such as, temporal templates [Hauskrecht et al., 2010, Valko and Hauskrecht, 2010], predictive temporal patterns[Batal et al., 2011, Batal et al., 2012b, Batal et al., 2012a, Batal et al., 2016], probabilistic state-space models [Liu and Hauskrecht, 2015b, Liu et al., 2013, Liu and Hauskrecht, 2015a, Liu and Hauskrecht, 2016a, Liu and Hauskrecht, 2016b, Liu and Hauskrecht, 2017], or modern deep learning methods based on RNNs [Lee and Hauskrecht, 2019, Lee and Hauskrecht, 2020], but none of these is human friendly and can be immediately used either for human instance review or for explaining the model to the human expert. An interesting open research question is how to utilize or transform these efficient machine learning representations and their ability to summarize complex data instances, also to support the human review and case assessment.

Bibliography

[Batal et al., 2016] Batal, I., Cooper, G. F., Fradkin, D., Harrison, J., Moerchen, F., and Hauskrecht, M. (2016). An efficient pattern mining approach for event detection in multivariate temporal data. *Knowledge and information systems*, 46(1):115–150.

[Batal et al., 2012a] Batal, I., Fradkin, D., Harrison, J., Moerchen, F., and Hauskrecht, M. (2012a). Mining Recent Temporal Patterns for Event Detection in Multivariate Time Series Data. In *Proceedings of the international conference on Knowledge Discovery and Data mining (SIGKDD)*.

[Batal et al., 2013] Batal, I., Hong, C., and Hauskrecht, M. (2013). An efficient probabilistic framework for multi-dimensional classification. pages 2417–2422.

[Batal et al., 2011] Batal, I., Valizadegan, H., Cooper, G. F., and Hauskrecht, M. (2011). A pattern mining approach for classifying multivariate temporal data. In *Proceedings of the IEEE international conference on bioinformatics and biomedicine (BIBM)*.

[Batal et al., 2012b] Batal, I., Valizadegan, H., Cooper, G. F., and Hauskrecht, M. (2012b). A Temporal Pattern Mining Approach for Classifying Electronic Health Record Data. *ACM Transaction on Intelligent Systems and Technology (ACM TIST), Special Issue on Health Informatics*.

[Bertin-Mahieux et al., 2011] Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

[Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144C152, New York, NY, USA. Association for Computing Machinery.

[Bottou and Bousquet, 2008] Bottou, L. and Bousquet, O. (2008). The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, volume 20, pages 161–168.

[Boutell et al., 2004] Boutell, M., Luo, J., Shen, X., and Brown, C. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37:1757–1771.

[Bradley and Guestrin, 2010] Bradley, J. K. and Guestrin, C. (2010). Learning tree conditional random fields. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 127–134, USA. Omnipress.

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

[Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

[Bucak et al., 2009] Bucak, S. S., Mallapragada, P. K., Jin, R., and Jain, A. K. (2009). Efficient multi-label ranking for multi-class learning: Application to object recognition. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2098–2105.

[Burbidge et al., 2007] Burbidge, R., Rowland, J. J., and King, R. D. (2007). Active learning for regression based on query by committee. In Yin, H., Tino, P., Corchado, E., Byrne, W., and Yao, X., editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, pages 209–218, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Chen et al., 2017] Chen, W., Chen, X., Zhang, J., and Huang, K. (2017). Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412.

[Chu and Keerthi, 2005] Chu, W. and Keerthi, S. S. (2005). New approaches to support vector ordinal regression. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 145–152, New York, NY, USA. ACM.

[Clare and King, 2001] Clare, A. and King, R. D. (2001). Knowledge discovery in multi-label phenotype data. In De Raedt, L. and Siebes, A., editors, *Principles of Data Mining and Knowledge Discovery*, pages 42–53, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Cohn et al., 1996] Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145.

[Collins, 2003] Collins, M. (2003). Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

[Culotta and McCallum, 2005] Culotta, A. and McCallum, A. (2005). Reducing labeling effort for structured prediction tasks. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI '05, page 746C751. AAAI Press.

[Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314.

[Domingos and Pazzani, 1997] Domingos, P. and Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130.

[Druck et al., 2009] Druck, G., Settles, B., and McCallum, A. (2009). Active learning by labeling features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, page 81C90, USA. Association for Computational Linguistics.

[Fisher, 1936] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188.

[Freedman and Diaconis, 1981] Freedman, D. and Diaconis, P. (1981). On the histogram as a density estimator. *Probability Theory and Related Fields*, 57(4):453–476.

[Friedman, 2002] Friedman, J. H. (2002). Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38(4):367C378.

[Geman et al., 1992] Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58.

[Griffin and Tversky, 1992] Griffin, D. and Tversky, A. (1992). The weighing of evidence and the determinants of confidence. *Cognitive Psychology*, 24(3):411–435.

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.

[Hauskrecht et al., 2016] Hauskrecht, M., Batal, I., Hong, C., Nguyen, Q., Cooper, G. F., Visweswaran, S., and Clermont, G. (2016). Outlier-based detection of unusual patient-management actions: an icu study. *Journal of biomedical informatics*, 64:211–221.

[Hauskrecht et al., 2013] Hauskrecht, M., Batal, I., Valko, M., Visweswaran, S., Cooper, G. F., and Clermont, G. (2013). Outlier detection for patient monitoring and alerting. *Journal of biomedical informatics*, 46(1):47–55.

[Hauskrecht et al., 2005] Hauskrecht, M., Pelikan, R., Malehorn, D. E., Bigbee, W. L., Lotze, M. T., Zeh, H. J., Whitcomb, D. C., and Lyons-Weiler, J. (2005). Feature Selection for Classification of SELDI-TOF-MS Proteomic Profiles. *Applied Bioinformatics*, 4(4):227–246.

[Hauskrecht et al., 2007] Hauskrecht, M., Pelikan, R., Valko, M., and Lyons-Weiler, J. (2007). *Feature Selection and Dimensionality Reduction in Genomics and Proteomics*, pages 149–172. Springer.

[Hauskrecht et al., 2010] Hauskrecht, M., Valko, M., Batal, I., Clermont, G., Visweswaram, S., and Cooper, G. (2010). Conditional Outlier Detection for Clinical Alerting. In *Proceedings of the American Medical Informatics Association (AMIA)*.

[Haussler, 1989] Haussler, D. (1989). Learning conjunctive concepts in structural domains. *Mach. Learn.*, 4(1):7C40.

[He et al., 2012] He, X., Wang, Z., Jin, C., Zheng, Y., and Xue, X. (2012). A simplified multi-class support vector machine with reduced dual optimization. *Pattern Recognition Letters*, 33(1):71–82.

[Heim et al., 2015] Heim, E., Berger, M., Seversky, L. M., and Hauskrecht, M. (2015). Efficient online relative comparison kernel learning. *CoRR*, abs/1501.01242.

[Heim and Hauskrecht, 2015] Heim, E. and Hauskrecht, M. (2015). Sparse multidimensional patient modeling using auxiliary confidence labels. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 331–336.

[Heim et al., 2014] Heim, E., Valizadegan, H., and Hauskrecht, M. (2014). Relative comparison kernel learning with auxiliary kernels. In Calders, T., Esposito, F., Hüllermeier, E., and Meo, R., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 563–578, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Herbrich et al., 1999] Herbrich, R., Graepel, T., and Obermayer, K. (1999). Support vector learning for ordinal regression. In *International Conference on Artificial Neural Networks*, pages 97–102.

[Ho, 1995] Ho, T. K. (1995). Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, page 278, USA. IEEE Computer Society.

[Ho, 1998] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832C844.

[Hong et al., 2014] Hong, C., Batal, I., and Hauskrecht, M. (2014). A mixtures-of-trees framework for multi-label classification. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, page 211C220, New York, NY, USA. Association for Computing Machinery.

[Hong et al., 2015] Hong, C., Batal, I., and Hauskrecht, M. (2015). A generalized mixture framework for multi-label classification. *Proceedings of the ... SIAM International Conference on Data Mining. SIAM International Conference on Data Mining*, 2015:712–720.

[Hwa, 2001] Hwa, R. (2001). On minimizing training corpus for parser acquisition. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL)*.

[Järvelin and Kekäläinen, 2002] Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

[Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, page 137C142, Berlin, Heidelberg. Springer-Verlag.

[Joachims, 2002] Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142.

[Jung and Tewari, 2018] Jung, Y. and Tewari, A. (2018). Online boosting algorithms for multi-label ranking. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[Juslin et al., 1998] Juslin, P., Olsson, H., and Winman, A. (1998). The calibration issue: Theoretical comments on suantak, bolger, and ferrell. *Organizational Behavior and Human Decision Processes*, 73(1):3–26.

[Lafferty et al., 2001] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Lee and Hauskrecht, 2019] Lee, J. M. and Hauskrecht, M. (2019). Recent-context-aware LSTM-based Clinical Time-Series Prediction. In *In Proceedings of AI in Medicine Europe (AIME)*.

[Lee and Hauskrecht, 2020] Lee, J. M. and Hauskrecht, M. (2020). Multi-scale temporal memory for clinical event time-series prediction. In *In Proceedings of the International Conference on AI in Medicine (AIME)*.

[Lewis and Gale, 1994] Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin, IE. Springer Verlag, Heidelberg, DE.

[Likert, 1932] Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55.

[Liu and Hauskrecht, 2019] Liu, S. and Hauskrecht, M. (2019). Nonparametric regressive point processes based on conditional gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1062–1072.

[Liu, 2009] Liu, T.-Y. (2009). Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225C331.

[Liu and Hauskrecht, 2015a] Liu, Z. and Hauskrecht, M. (2015a). Clinical time series prediction: Toward a hierarchical dynamical system framework. *Artificial Intelligence in Medicine*, 65(1):5–18.

[Liu and Hauskrecht, 2015b] Liu, Z. and Hauskrecht, M. (2015b). A regularized linear dynamical system framework for multivariate time series analysis. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[Liu and Hauskrecht, 2016a] Liu, Z. and Hauskrecht, M. (2016a). Learning adaptive forecasting models from irregularly sampled multivariate clinical data. In *The 30th AAAI Conference on Artificial Intelligence*, pages 1273–1279.

[Liu and Hauskrecht, 2016b] Liu, Z. and Hauskrecht, M. (2016b). Learning linear dynamical systems from multivariate time series: A matrix factorization based framework. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 810–818. SIAM.

[Liu and Hauskrecht, 2017] Liu, Z. and Hauskrecht, M. (2017). A personalized predictive framework for multivariate clinical time series via adaptive model selection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1169–1177.

[Liu et al., 2013] Liu, Z., Wu, L., and Hauskrecht, M. (2013). Modeling clinical time series using gaussian process sequences. In *SIAM International Conference on Data Mining*.

[Luo and Hauskrecht, 2018a] Luo, Z. and Hauskrecht, M. (2018a). Hierarchical active learning with group proportion feedback. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI'18*, pages 2532–2538.

[Luo and Hauskrecht, 2018b] Luo, Z. and Hauskrecht, M. (2018b). Hierarchical active learning with proportion feedback on regions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 464–480. Springer.

[Luo and Hauskrecht, 2019] Luo, Z. and Hauskrecht, M. (2019). Region-based active learning with hierarchical and adaptive region construction. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 441–449. SIAM.

[Luo and Hauskrecht, 2020] Luo, Z. and Hauskrecht, M. (2020). Hierarchical active learning with overlapping regions. In *Proceedings of the 2020 ACM International Conferences on Information and Knowledge Management*.

[Mason et al., 2000] Mason, L., Baxter, J., Bartlett, P., and Frean, M. (2000). Boosting algorithms as gradient descent. In *In Advances in Neural Information Processing Systems 12*, pages 512–518. MIT Press.

[McCullagh and Nelder, 1989] McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models (Second edition)*. London: Chapman & Hall.

[Mitchell, 1979] Mitchell, T. M. (1979). An analysis of generalization as a search problem. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI79, page 577C582, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Mohri et al., 2012] Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of Machine Learning*. The MIT Press.

[Mozafari et al., 2012] Mozafari, B., Sarkar, P., Franklin, M. J., Jordan, M. I., and Madden, S. (2012). Active learning for crowd-sourced databases. *CoRR*, abs/1209.3686.

[Naeini et al., 2015] Naeini, M. P., Cooper, G. F., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*.

[Nguyen et al., 2011a] Nguyen, Q., Valizadegan, H., and Hauskrecht, M. (2011a). Learning classification with auxiliary probabilistic information. In *IEEE International Conference on Data Mining*, pages 477–486.

[Nguyen et al., 2011b] Nguyen, Q., Valizadegan, H., and Hauskrecht, M. (2011b). Sample-efficient learning with auxiliary class-label information. In *Proceedings of the Annual American Medical Informatics Association Symposium*, pages 1004–1012.

[Nguyen et al., 2013] Nguyen, Q., Valizadegan, H., and Hauskrecht, M. (2013). Learning classification models with soft-label information. *Journal of American Medical Informatics Association*.

[Peng and Wong, 2014] Peng, P. and Wong, R. C.-W. (2014). Selective sampling on probabilistic data. In *SIAM International Conference on Data Mining*, pages 28–36.

[Peng et al., 2014] Peng, P., Wong, R. C.-W., and Yu, P. S. (2014). Learning on probabilistic labels. In *SIAM International Conference on Data Mining*, pages 307–315.

[Platt, 1999] Platt, J. C. (1999). Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, Cambridge, MA, USA.

[Poggio and Cauwenberghs, 2001] Poggio, T. and Cauwenberghs, G. (2001). Incremental and decremental support vector machine learning. In *Advances in Neural information processing systems*, volume 13, page 409.

[Radlinski and Joachims, 2005] Radlinski, F. and Joachims, T. (2005). Query chains: Learning to rank from implicit feedback. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, page 239C248, New York, NY, USA. Association for Computing Machinery.

[Read et al., 2009] Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2009). Classifier chains for multi-label classification. In Buntine, W., Grobelnik, M., Mladenić, D., and Shawe-Taylor, J., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 254–269, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Roy and McCallum, 2001] Roy, N. and McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In Brodley, C. E. and Danyluk, A. P., editors, *Proceedings of the 18th International Conferenceon on Machine Learning*, pages 441–448, Williams College,Williamstown, MA, USA. Morgan Kaufmann.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). *Learning Internal Representations by Error Propagation*, page 318C362. MIT Press, Cambridge, MA, USA.

[Schapire, 1990] Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5(2):197–227.

[Scheffer et al., 2001] Scheffer, T., Decomain, C., and Wrobel, S. (2001). Active hidden markov models for information extraction. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, IDA '01, page 309C318, Berlin, Heidelberg. Springer-Verlag.

[Settles, 2010] Settles, B. (2010). Active learning literature survey. Technical report.

[Settles et al., 2008a] Settles, B., Craven, M., and Friedland, L. (2008a). Active learning with real annotation costs. In *Proceedings of the NIPS workshop on cost-sensitive learning*, pages 1–10.

[Settles et al., 2008b] Settles, B., Craven, M., and Ray, S. (2008b). Multiple-instance active learning. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in neural information processing systems*, pages 1289–1296. MIT Press.

[Seung et al., 1992] Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 287–294, Pittsburgh, Pennsylvania. ACM Press.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423.

[Theodoridis and Koutroumbas, 2008] Theodoridis, S. and Koutroumbas, K. (2008). *Pattern Recognition, Fourth Edition*. Academic Press, Inc., USA, 4th edition.

[Thiel, 2008] Thiel, C. (2008). *Classification on Soft Labels Is Robust against Label Noise*, pages 65–73. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Tong and Koller, 2000] Tong, S. and Koller, D. (2000). Active learning for parameter estimation in bayesian networks. In *Advances in Neural Information Processing Systems*, pages 647–653. MIT Press.

[Tosh and Dasgupta, 2018] Tosh, C. and Dasgupta, S. (2018). Structural query-by-committee. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018)*.

[Tsoumakas et al., 2010] Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). *Mining Multi-label Data*, pages 667–685. Springer US, Boston, MA.

[Tung, 2009] Tung, A. K. H. (2009). *Rule-based Classification*, pages 2459–2462. Springer US, Boston, MA.

[Valizadegan et al., 2012] Valizadegan, H., Nguyen, Q., and Hauskrecht, M. (2012). Learning medical diagnosis models from multiple experts. In *Proceedings of the Annual American Medical Informatics Association*, pages 921–30.

[Valizadegan et al., 2013] Valizadegan, H., Nguyen, Q., and Hauskrecht, M. (2013). Learning classification models from multiple experts. *Journal of Biomedical Informatics*, pages 1125–1135.

[Valko and Hauskrecht, 2010] Valko, M. and Hauskrecht, M. (2010). Feature importance analysis for patient management decisions. In *Proceedings of the 13th International Congress on Medical Informatics*, pages 861–865.

[Van Der Malsburg, 1986] Van Der Malsburg, C. (1986). Frank rosenblatt: Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. In Palm, G. and Aertsen, A., editors, *Brain Theory*, pages 245–248, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Vapnik, 1995] Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, New-York.

[Vembu and Gärtner, 2011] Vembu, S. and Gärtner, T. (2011). *Label Ranking Algorithms: A Survey*, pages 45–64. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Weston et al., 1999] Weston, J., Gammerman, A., Stitson, M., Vapnik, V., Vovk, V., and Watkins, C. (1999). Support vector density estimation. *Advances in Kernel MethodsłSupport Vector Learning*, pages 293–306.

[Xu et al., 2018] Xu, N., Tao, A., and Geng, X. (2018). Label enhancement for label distribution learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*.

[Xue and Hauskrecht, 2017a] Xue, Y. and Hauskrecht, M. (2017a). Active learning of classification models with likert-scale feedback. In *SIAM International Conference on Data Mining*, pages 28–35.

[Xue and Hauskrecht, 2017b] Xue, Y. and Hauskrecht, M. (2017b). Efficient learning of classification models from soft-label information by binning and ranking. In *Proceedings of the 30th International Florida AI Research Society Conference*, pages 164–169.

[Xue and Hauskrecht, 2018] Xue, Y. and Hauskrecht, M. (2018). Active learning of multi-class classifiers with auxiliary probabilistic information. In *Proceedings of the 31st International Florida AI Research Society Conference.*

[Xue and Hauskrecht, 2019] Xue, Y. and Hauskrecht, M. (2019). Active learning of multi-class classification models from ordered class sets. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*.

[Zhai et al., 2019] Zhai, Y., Guo, X., Lu, Y., and Li, H. (2019). In defense of the classification loss for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

[Zhao et al., 2009] Zhao, S., Tsang, E. C., Chen, D., and Wang, X. (2009). Building a rule-based classifierła fuzzy-rough set approach. *IEEE Transactions on Knowledge and Data Engineering*, 22(5):624–638.

[Zhou et al., 2014] Zhou, Y., Liu, Y., Yang, J., He, X., and Liu, L. (2014). A taxonomy of label ranking algorithms. *Journal of Computers*, 9.

[Zhou and Zhang, 2002] Zhou, Z.-H. and Zhang, M.-L. (2002). Neural networks for multi-instance learning. In *Proceedings of the International Conference on Intelligent Information Technology, Beijing, China*, pages 455–459.

[Zhu, 2005] Zhu, X. (2005). *Semi-supervised learning with graphs*. PhD thesis, Pittsburgh, PA, USA. AAI3179046.