TEAMOPT: Interactive Team Optimization in Big Networks

Liangyue Li
Arizona State University
liangyue@asu.edu

Hanghang Tong
Arizona State University
hanghang.tong@asu.edu

Nan Cao
NYU Shanghai
nan.cao@nyu.edu

Kate Ehrlich
IBM Research
katee@us.ibm.com

Yu-Ru Lin
University of Pittsburgh
yurulin@pitt.edu

Norbou Buchler
US Army Research Laboratory
norbou.buchler.civ@mail.mil

ABSTRACT

The science of team science is a rapidly emerging research field that studies strategies to understand and enhance the process and outcomes of collaborative, team-based research. An interesting research question we address in this work is how to maintain and optimize the team performance should certain changes happen to the team. In particular, we take the network approach to understanding the teams and consider optimizing the teams with several operations (e.g., replacement, expansion, shrinkage).

We develop TeamOpt, a system to assist users in optimizing the team performance interactively to support the changes to a team. TeamOpt takes as input a large network of individuals (e.g., co-author network of researchers) and is able to assist users in assembling a team with specific requirements and optimizing the team in response to the changes made to the team. It is effective in finding the best candidates, and interactive with users' feedback in the loop. The system is developed using HTML5, JavaScript, D3.js (front-end) and Python CGI (back-end). A prototype system is already deployed. We will invite the audience to experiment with our TeamOpt in terms of its effectiveness, efficiency and applicability to various scenarios.

1. INTRODUCTION

The science of team science is a rapidly emerging research field that "encompasses both conceptual and methodological strategies aimed at understanding and enhancing the process and outcomes of collaborative, team-based research" [6]. Both internal and external environments can drive the changes of a team, e.g., the churn of a team member, team expansion, team shrinkage. An interesting research question is how to maintain and optimize the team performance under such circumstances.

The network approach to understanding the teams places the individuals in their social context and treats a team as a subgraph embedded in a larger social network. In this work, we consider optimizing the team performance interactively in a large network and specifically address a family of problems under the scope of Team Enhancement, including (1) Team Replacement [4], i.e., to find a best candidate to replace a team member who is leaving; (2) Team Refinement, i.e., to refine the team by editing the skills and connectivity among the members; (3) Team Expansion, i.e., to find a best candidate to join the team; (4) Team Shrinkage, i.e., to find a proper team member to leave the team; and (5) Team Conflict Resolution, i.e., to resolve the conflict between two or more team members.

We develop TeamOpt (http://team-net-work.org/), a system to assist users in optimizing the team performance interactively to support the changes to a team. TeamOpt takes as input a large network of individuals (e.g., co-author network of researchers) and is able to assist users in assembling a team with specific requirements and optimizing the team in response to the changes made to the team. To the best of our knowledge, this is the first system specializing in forming and optimizing teams with the following key features. First (*effectiveness*), we carefully identify our design objectives and develop effective algorithms with the key technique of graph kernels. Compared with other competitors, our algorithm can achieve the highest precision and recall in finding the best team member candidate. Second (*interaction*), we design fast solutions to our algorithms, enabling an interactive user experience with users' feedback in the loop. Third (*deployment*), we build our system with HTML5, Javascript, D3.js (front-end) and Python CGI (back-end).

The rest of the paper is organized as follows. Section 2 demonstrates the main functionality of TeamOpt. Section 3 presents the technical details behind the scene. Section 4 reviews the related works and Section 5 concludes the paper.

2. FUNCTIONALITY DEMONSTRATION

In this section, we demonstrate the main functionalities of TeamOpt. Figure 1 shows the user interface of the system, where Figure 1(1) is the main component allowing the user to explore the network dataset, form a team and perform a series of operations (e.g., replacement, refinement, expansion) to optimize the team in terms of performance/output. Figure 1(2) visualizes the hierarchical clusters in an icicle tree view [2], exhibiting the overall hierarchy structure in the network. Figure 1(3) provides the overview of the parent cluster of the current one in the hierarchy and Figure 1(4) visualizes the candidate teams to support the team optimization operations. We detail the functionalities next.

Team Formation: the system provides users with three ways of forming a team to suit their different needs. (1)

CIKM'16 October 24 - November 28, 2016, Indianapolis, IN, USA

ACM ISBN 978-1-4503-4073-1/16/10.

DOI: http://dx.doi.org/10.1145/2983323.2983340

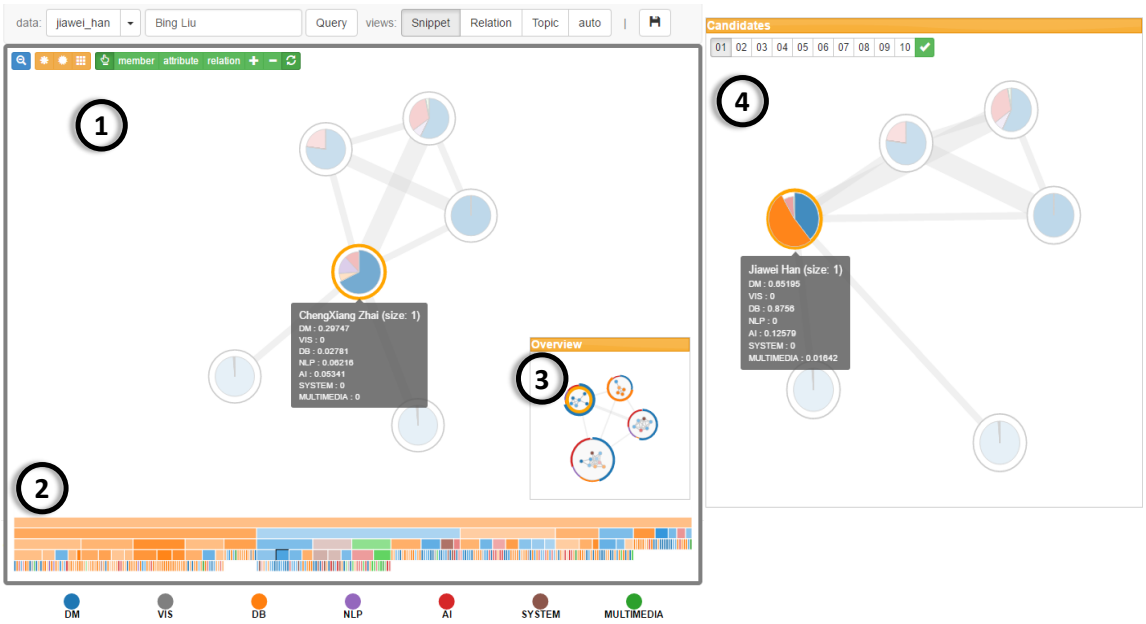

Figure 1: An illustrative example of using our TeamOpt to find a candidate (e.g., Jiawei Han) to replace one team member (e.g., ChengXiang Zhai) when the member has to leave the team.

Skill coverage: the user specifies a subset of skills required to perform a task and the system will output a set of individuals with the minimum communication cost that can cover these skills [3]. (2) *Template matching:* the user has a specific configuration of the team in terms of the skills each team member has as well as how the team members are connected to each other. The system would then return a set of team members from the network dataset that best matches the user-input configuration. (3) *Ego network:* the user is interested in having a particular person on the team and s/he can query the system with the member's name. The system would locate the member in the dataset and show the member's ego network, including the connections among the member's immediate neighbors.

Team Replacement: the churn of team members is a common phenomenon across a wide variety of application domains, including sports, business, entertainment, etc. Figure 1 shows an example of the system in assisting the user find a best candidate that can replace the member who is leaving. The user only needs to select the member that is leaving and the system would recommend a list of candidates with the new teams visualized for an easy visual comparison.

Team Refinement: most often than not, the user might not be satisfied with the initial team formed. In this case, the user can refine the team by further editing the skills of the members and/or connections between them. The system would then recommend a list of candidate teams that best fulfill the new requirements.

Team Expansion: sometimes a new team member is needed to ramp up the team's capacity. The user can indicate the skills the new member should have as well as the connections s/he has to the current team members. The system would then recommend a list of candidates that best suit the user's requirement.

Team Shrinkage: on the contrary to the team expansion, in team shrinkage, the size of a team needs to be reduced in response to new challenges such as a shortage of the available resources. In this case, the system would recommend a list of candidates to leave the team without exerting too much negative effect on the team performance.

Team Conflict Resolution: it happens that two or multiple team members have conflict of interest with each other, in which case, the system can resolve such conflict by either replacing one conflicting member or asking s/he to leave the team.

The system is developed using HTML5, JavaScript, D3.js (front-end) and Python CGI (back-end). A prototype system is deployed at http://team-net-work.org/.

3. TECHNICAL DETAILS

In this section, we first present the algorithm that enables the team optimization operations in TeamOpt, followed by some evaluation results.

3.1 Algorithm Details

We first show the solution for Team Replacement and later point out that other operations (e.g., Team Refinement, Team Expansion) can be solved in a strategically similar way. The churn of a member might adversely affect the performance of a team for several reasons, including loss of certain skills, change of the network connectivity, and dissolution of certain critical skill configurations (e.g., the *triangle offense* created by *the center, the forward,* and *the guard* in NBA teams sustains as a classic tactic). As a result, our goal in finding the best candidate for the replacement is to minimize such negative disruptions and maintain the stability of the team as an organic system of skills and network connectivities. Realizing this, we devote ourselves to finding the candidates that are as similar to the member leaving as possible and propose the concept of *team context aware similarity* to measure the similarity of the two members in the context of the team. Such similarity is able to capture the following three aspects:

- *skill matching:* the new member should have a similar

skill set to the person leaving to preserve the skill set of the whole team.

- *structural matching:* the new member should have a similar network connectivity to the rest of the team members.
- *skill configurations matching:* the new member should have similar skill configurations among the sub-groups in the team required by certain sub-tasks of the team.

Mathematically, the *team context aware similarity* can be instantiated by the attributed graph kernel defined on the current and new teams. Formally, the best candidate for Team Replacement can be described by the following objective:

$$q = \arg\max_{j \notin \mathcal{T}} \mathrm{Ker}(G(\mathcal{T}), G(\mathcal{T}_{p \to j})), \quad (1)$$

where the individual q is to replace the member p, $\mathcal{T}$ is used to index the members of a team, $G(\mathcal{T})$ is the induced subgraph by the current team members from the attributed network, and $G(\mathcal{T}_{p \to j})$ is the subgraph induced after the team member p is replaced by another individual j in the network. The function $\mathrm{Ker}(\cdot)$ is the kernel between the two attributed subgraphs. The basic idea behind graph kernels is to aggregate the similarities between the subgraphs in the two graphs, which corresponds to skill configurations among the sub-groups in the two teams. As such, the graph kernel can naturally encode the three aspects required by the *team context aware similarity*. We adopt the random walk based graph kernel [5] in the system implementation for its mathematical elegance and superior empirical performance. The solution to other operations are similar, with the key idea of having a new team (e.g., the team after expansion, refinement, and shrinkage) most similar to the original team. The similarity is still measured using graph kernels.

To enable an interactive user experience, we need to address the computational challenges brought up by the large-scale networks. To scale up, we first design an efficient pruning strategy to filter out those unpromising candidates, i.e., those who do not have any connections to any of the rest team members. We then speed up each graph kernel computation by exploring the smoothness and correspondences between the existing and the new teams.

3.2 Evaluations

We perform a user study with 20 people aged from 22 to 35. We choose 10 papers from various fields, replace one author of each paper, run our method and the two comparison methods, and each of them recommends five candidates. We then mix the 15 recommendations and ask the users to (a) mark exactly one best replacement; and (b) mark all the good replacements. The result is shown in Figure 2 and our method is the best in terms of both precision and recall.

4. RELATED WORK

In data mining community, the pioneering work in assembling a team of experts in social networks requires the members in the team not only meet the skill requirements of the task, but also have the minimum communication cost [3]. Going beyond team formation, a recent work [4] designs fast algorithms for team member recommendation in response to the churn of a team member. An interactive system called "g-Miner" [1] is developed for mining groups with complex criteria.

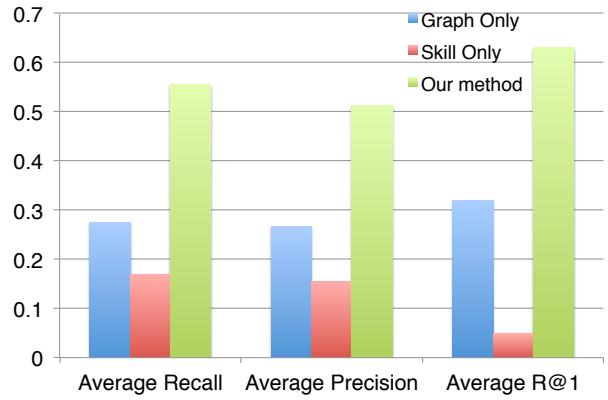

Figure 2: The comparison result of the user study.

5. CONCLUSIONS

In this work, we address the problem of maintaining and optimizing the team performance in response to certain changes made to the team. In particular, we treat a team as a subgraph embedded in a larger social network. We develop TeamOpt, a system to assist users in optimizing the team performance interactively to support the changes to a team. TeamOpt takes as input a large network of individuals (e.g., co-author network of researchers) and is able to assist users in assembling a team with specific requirements and optimizing the team in response to the changes. The system is effective in recommending the best candidates, and efficient to support interactive user experience. A prototype system is already deployed and we will invite the audience to play with the system for various application scenarios.

6. ACKNOWLEDGMENTS

This work is partially supported by the National Science Foundation under Grant No. IIS1017415, by DTRA under the grant number HDTRA1-16-0017, by Army Research Office under the contract number W911NF-16-1-0168, by National Institutes of Health under the grant number R01LM011986, Region II University Transportation Center under the project number 49997-33 25 and a Baidu gift.

7. REFERENCES

[1] N. Cao, Y. Lin, L. Li, and H. Tong. g-miner: Interactive visual group mining on multivariate graphs. In *CHI*, pages 279–288, 2015.

[2] J. B. Kruskal and J. M. Landwehr. Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, 37(2):162–168, 1983.

[3] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *KDD*, pages 467–476. ACM, 2009.

[4] L. Li, H. Tong, N. Cao, K. Ehrlich, Y. Lin, and N. Buchler. Replacing the irreplaceable: Fast algorithms for team member recommendation. In *WWW*, pages 636–646, 2015.

[5] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *The Journal of Machine Learning Research*, 11:1201–1242, 2010.

[6] Wikipedia. Science of team science. https://en.wikipedia.org/wiki/Science_of_team_science.