Constrained Multi-Aspect Expertise Matching for Committee Review Assignment

Maryam Karimzadehgan
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
mkarimz2@illinoisedu

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
czhai@cs.illinois.edu

ABSTRACT

Automatic review assignment can significantly improve the productivity of many people such as conference organizers, journal editors and grant administrators. Most previous works have set the problem up as using a paper as a query to *independently* "retrieve" a set of reviewers that should review the paper. A more appropriate formulation of the problem would be to simultaneously optimize the assignments of all the papers to an entire committee of reviewers under constraints such as the review quota. In this paper, we solve the problem of committee review assignment with multi-aspect expertise matching by casting it as an integer linear programming problem. The proposed algorithm can naturally accommodate any probabilistic or deterministic method for modeling multiple aspects to automate committee review assignments. Evaluation using an existing data set shows that the proposed algorithm is effective for committee review assignments based on multi-aspect expertise matching.

Categories and Subject Descriptors: H.3.3 Information Search and Retrieval: Retrieval models

General Terms: Algorithms, Experimentation.

Keywords: Topic Models, Review Assignment, Algorithms, Combinatorial Optimization.

1. INTRODUCTION

Automatic review assignment is very beneficial for many people such as conference organizers, journal editors, and grant administrators. In the review assignment task, matching a set of candidate reviewers with a paper to be reviewed is needed. Manually assigning reviewers to papers as is done in some conferences is known to be quite time-consuming.

In most studies of review assignment (e.g., [1, 3, 4, 6, 10]), the problem is considered as a retrieval problem, where the query is a paper (or a grant proposal) to be reviewed and a candidate reviewer is represented as a text document. One main drawback of all these works is that a paper or proposal is matched as a *whole* component without taking into account the multiple subtopics.[1]

In our previous work [8], we have studied how to match reviewers with papers based on subtopics and the proposed methods are shown to increase the aspect coverage for automatic review assignment. However, in this work and a lot of other previous works, the assignment of reviewers to each paper is done *independently* without considering the whole committee. This makes it hard to balance the review load among a set of reviewers, and may result in assigning too many papers to a reviewer with expertise on a popular topic. To balance the review load and conform to the review quota of each reviewer, it is necessary to set up the problem as to simultaneously assign papers to all the reviewers on a committee with consideration of review-load balancing. We call this problem Committee Review Assignment (CRA). Although the CRA problem has been previously studied by a few researchers [2, 9, 5, 11], none of them has considered multiple aspects of topics and expertise in matching papers with reviewers.

In this paper, we study a novel setup of the CRA problem where the goal is to assign a pool of reviewers on a committee to a set of papers based on multi-aspect expertise matching (i.e., the assigned reviewers should cover as many subtopics of the paper as possible) and with constraints of review quota for reviewers. We call this problem Constrained Multi-Aspect Committee Review Assignment (CMACRA). We propose to solve the CMACRA problem by casting it as an integer programming problem. In our optimization setup, matching of reviewers with a paper is done based on matching of multiple aspects of expertise. The preferences and requirements are captured through a set of constraints in the integer programming formulation, and the objective function maximizes average coverage of multiple aspects. The proposed algorithm is quite general; it allows us to set a potentially different review quota for each reviewer and can naturally accommodate any probabilistic or deterministic method for modeling multiple aspects to automate committee review assignments.

To evaluate the effectiveness of our algorithm, we use the measures and gold standard data in our previous work [8] [2]. Our experiment results show that the proposed committee review assignment algorithm is quite effective for the CMACRA task and outperforms a heuristic greedy algo-

[1] Aspect and subtopic will be used interchangeably throughout the paper.

[2] Available at http://timan.cs.uiuc.edu/data/review.html.

CIKM'09, November 2–6, 2009, Hong Kong, China.

rithm for assignment.

2. CONSTRAINED MULTI-ASPECT COMMITTEE REVIEW ASSIGNMENT

Informally, the problem of Constrained Multi-Aspect Committee Review Assignment (CMACRA) is to reflect a very common application scenario such as conference review assignment where the goal is to assign a set of reviewers to a set of papers so that (1) each paper will be reviewed by a certain number of reviewers; (2) each reviewer would not review more than a specified number of papers; (3) the reviewers assigned to a paper have the expertise to review the paper; and (4) the combined expertise of the reviewers assigned to a paper would cover well all the subtopics of the paper.

As a computation problem, CMACRA takes the following information as the input:

- A set of n papers: $\mathcal{P} = \{p_1, ..., p_n\}$ where each p_i is a paper.
- A set of m reviewers: $\mathcal{R} = \{r_1, ..., r_m\}$ where each r_i is a reviewer.
- A set of reviewer quota limits: $NR = \{NR_1, ..., NR_m\}$ where NR_i is the maximum number of papers a reviewer r_i can review.
- A set of numbers of reviewers to be assigned to a paper: $NP = \{NP_1, ..., NP_n\}$ where NP_i is the number of reviewers that should be assigned to paper p_i.

And the output is a set of assignments of reviewers to papers, which can be represented as an $n \times m$ matrix M with $M_{ij} \in \{0, 1\}$ indicating whether reviewer r_i is assigned to paper p_j. ($M_{ij} = 1$ means that reviewer r_i has been assigned to review paper p_j.)

To respect the reviewer quota limits and to ensure that each paper gets the right number of reviewers, we require M to satisfy the following two constraints:

$$\begin{array}{l}(1)\ \forall i \in [1, m], \sum_{j=1}^{n} M_{ij} \leq NR_i \\ (2)\ \forall j \in [1, n], \sum_{i=1}^{m} M_{ij} = NP_j\end{array}$$

Naturally, we assume that there are sufficient reviewers to review all the papers subject to the quota constraints. That is, $\sum_{j=1}^{n} NP_j \leq \sum_{i=1}^{m} NR_i$.

In addition, we would also like the review assignments to match the expertise of the assigned reviewers with the topic of the paper well, and ideally, the reviewers can cover all the subtopics of the paper. Formally, let $\tau = (\tau_1, ..., \tau_k)$ be a set of k subtopics that can characterize the content of a paper as well as the expertise of a reviewer, and τ_i is a specific topic. These subtopics can be either from the list of topic keywords that are typically provided in a conference management system to facilitate review assignments or automatically learned via statistical topic models such as Probabilistic Latent Semantic Indexing (PLSA) [7] from the publications of reviewers as done in our previous work [8]. The subtopics in the first case are usually designed by human experts that run a conference such as program chairs, and both the authors and reviewers would be asked to choose some specific keywords to describe the content of the paper and the expertise of the reviewer, respectively. Thus we would have access to *deterministic* assignments of subtopics to the papers and reviewers. In the second, a subtopic can be characterized by a word distribution, and in general, a paper and a reviewer would get a *probabilistic* assignment of subtopics to characterize the content of the paper and the expertise of the reviewer.

Thus we assume that we have two matrices P and R available, which represent our knowledge about the subtopics of the content of a paper and the subtopics of the expertise of a reviewer, respectively. P is a $n \times k$ matrix where P_{ij} is a probability (or any positive weight) indicating how likely subtopic τ_j represents the content of paper p_i. R is a $m \times k$ matrix where R_{ij} is a probability (or any weight) indicating how likely subtopic τ_j represents the expertise of reviewer r_i. Clearly, when P_{ij} and R_{ij} take binary values, we would end up having deterministic assignments of subtopics to papers and reviewers.

3. ALGORITHMS FOR CMACRA

Our main idea for solving the CMACRA problem is to cast it as a tractable optimization problem, i.e., an integer linear programming problem. We also present a heuristic greedy algorithm as our baseline algorithm that only works for the scenario of deterministic subtopic assignments to papers and reviewers.

3.1 A greedy algorithm

In this algorithm, we would optimize the review assignments for each paper iteratively. The algorithm only works for the scenario of deterministic assignments of topics to papers and reviewers. It works as follows:

First, the papers are decreasingly sorted according to the number of subtopics they contain, i.e., the paper with the largest number of subtopics is ranked first. We then start off with this ranked list of the papers. At each assignment stage, the best reviewer that can cover most subtopics of the paper is assigned. In addition, the review quota and paper quota are checked, i.e., the number of papers assigned to each reviewer and the number of reviewers assigned to each paper. If the review quota is reached, that reviewer is removed from our reviewer pool; the same is done when the paper quota is satisfied. This process is repeated until reviewers are assigned to all the papers.

3.2 An integer linear programming algorithm

In our formal definition of the CMACRA problem in Section 2, we have already naturally introduced several constraints, and the problem can be cast as an optimization problem where we seek an optimized assignment matrix M that would satisfy all the constraints as well as optimize the multi-aspect matching of expertise of reviewers and the content of each paper. Thus M_{ij} would then naturally become variables in the definition of the ILP problem. We need to introduce auxiliary variables to connect M_{ij} with subtopic assignments. We propose to introduce the following set of auxiliary variables:$\{t_{ij}\}_{i \in [1,n], j \in [1,k]}$, where $t_{ij} \in [0, NP_i]$ is an integer indicating the number of assigned reviewers that can cover subtopic τ_j for paper p_i. This allows us to define the following linear objective function to maximize:

$$Maximize(\sum_{i=1}^{n} \sum_{j=1}^{k} t_{ij})$$

Now we still need to connect t_{ij} with the review assignment matrix M. If the subtopic assignment is completely binary, which means that the element values of both matrices P and R are binary, it is relatively easy to see that we should have the following set of n inequality linear constraints, each for a paper:

$$\forall i \in [1,n], j \in [1,k], P_{ij}t_{ij} \leq \sum_{l=1}^{m} R_{lj}M_{li}$$

For paper p_i, this constraint says that if the paper covers subtopic j (i.e., $P_{ij} = 1$), t_{ij} can be as large as the actual number of reviewers assigned to paper p_i that can cover subtopic j. (Note that $R_{lj} = 1$ iff the expertise of reviewer r_l covers subtopic τ_j, and $M_{li} = 1$ iff reviewer r_l is assigned to paper i.)

If our subtopic assignment is probabilistic or fuzzy, the element values of P and R can be any positive real numbers. It turns out that the inequality above for t_{ij} would still make sense, though our solution would unlikely satisfy the equality. Specifically, the right hand side of the inequality is regarded to compute the weighted combined coverage of subtopic τ_j by all the assigned reviewers according to M, thus it still serves as a meaningful upper-bound. Similarly, the left hand side can also be interpreted as the desired coverage of subtopic τ_j since if P_{ij} is large, it would mean that paper p_i is very much likely about subtopic τ_j, and thus we would demand more coverage about τ_j.

Adding additional constraints introduced in Section 2, the complete ILP formulation of the CMACRA problem is shown in figure 1.

$$Maximize(\sum_{i=1}^{n}\sum_{j=1}^{k} t_{ij})$$

Subject to constraints:
C1: $\forall i \in [1,m], j \in [1,n]$, $M_{ij} \in \{0,1\}$
C2: $\forall i \in [1,n], j \in [1,k]$, $t_{ij} \in \{0,\ldots,NP_i\}$
C3: $\forall j \in [1,n]$, $\sum_{i=1}^{m} M_{ij} = NP_j$
C4: $\forall i \in [1,m]$, $\sum_{j=1}^{n} M_{ij} \leq NR_i$
C5: $\forall i \in [1,n], j \in [1,k]$ $P_{ij}t_{ij} \leq \sum_{l=1}^{m} R_{lj}M_{li}$

Figure 1: ILP formulation

If we have knowledge about conflict of interest of reviewers, we may further add the following additional constraint:

C6: $M_{ij} = 0$, if reviewer r_i has conflict of interest with paper p_j.

Our objective function indicates that for each paper we want to maximize both the number of covered topics and the number of reviewers that can cover each topic in the paper. Constraint **C1** shows that each variable is either one or zero where one means reviewer r_i is assigned to paper p_j. Constraint **C2** indicates that t_{ij} is an integer with minimum zero and maximum NP_i, where NP_i is the number of reviewers that should be assigned to paper p_i. Constraint **C3** indicates that each paper p_j will be assigned precisely NP_j reviewers. Constraint **C4** indicates that each reviewer r_i can review up to NR_i papers. Finally, constraint **C5** requires that variable t_{ij} be constrained by the actual coverage of subtopic τ_j by the assigned reviewers to paper p_i according to M.

3.3 Modeling and assigning subtopics

The proposed algorithms are based on the assumption that we have available a set of subtopics τ and the assignments of them to the papers and reviewers (i.e., P and R). This is a realistic assumption for a conference review system that requires all authors and reviewers to choose subtopic keywords, in which case we generally would have a binary P and R. In applications where we do not have such input from authors and reviewers, we may learn subtopics from the publications of reviewers and compute probabilistic assignments of subtopics to papers and reviewers as has been done in the previous work [8].

4. EXPERIMENT RESULTS

We use the same data sets and evaluation measures defined in our previous work [8].

4.1 Effectiveness of the ILP Algorithm

In this section, we compare the effectiveness of the ILP algorithm with the heuristic greedy algorithm. Since the greedy algorithm only works for the scenario of known subtopics, we use our gold standard data set to obtain subtopic assignments (i.e., matrices P and R).

In the first test, we vary the number of reviewers but the other parameters remain unchanged. The total number of topics are 25 (according to the gold standard data) and on average, reviewers' expertise topics are 5 (5 out of 25) and papers' topics are 3. Three reviewers are assigned to each paper and each reviewer gets up to 5 papers to review. Since there might be multiple *optimal solutions*, i.e., different assignments of reviewers to papers may lead to the same *optimal value* for the objective function for the ILP algorithm, we generate 10 such solutions and average over all. Figure 2 (left) shows the results of Average Confidence measure for the two algorithms with the error bars for different solutions (invisible error bars mean zero variance). Small error bars indeed indicate that the different solutions do not change the value for the evaluation measure. From the figure, we can see that as we increase the number of reviewers, the performance of both algorithms is getting better and the performance of the ILP algorithm is consistently much better than the greedy algorithm.

For the second test, we randomly select 30 (out of 189 in the gold standard data) reviewers to be assigned to 73 papers. In order to avoid bias, we repeat the sampling process for 10 times and get the average. The number of reviewers that can be assigned to each paper is 3 and we vary the number of papers that each reviewer can get. The results of Average Confidence measure are shown in figure 2 (middle). The figure shows the error bars for *10 different samples*. As we increase the number of papers that each reviewer can get, we are also increasing the resources, in terms of assigning good reviewers to many different papers. As a result, the performance of both algorithms becomes better. Again, comparing two algorithms shows that the ILP algorithm has a better performance than the greedy algorithm.

Finally, we test the performance of our algorithms when we have very limited resources, i.e., the maximum number of reviewers is 10 for 73 papers. Again we randomly select 10 reviewers and we repeat the sampling process for 10 times and get the average. Each paper gets 3 reviewers and the number of papers that each reviewer can get is calculated according to the number of reviewers that we have. For example, if we have 5 reviewers, each should get 44 papers. The results of Average Confidence measure are shown in figure 2 (right). As we increase the resources, i.e., the number of reviewers, the performance of both algorithms becomes better and and the ILP algorithm once again outperforms the greedy algorithm for all parameter values.

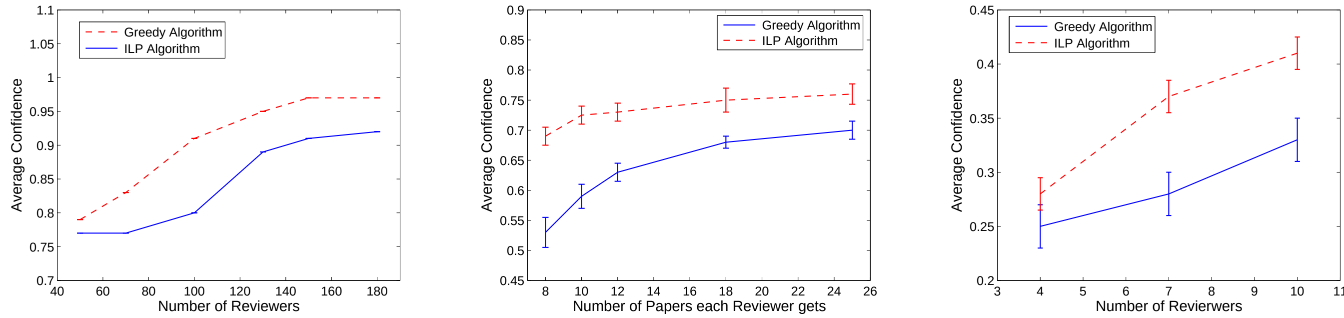

Figure 2: Comparison of ILP and greedy algorithms according to Average Confidence. In all cases, the number of papers is 73. Each paper gets 3 reviewers. The three figures show different variations. Left: vary the number of reviewers; each reviewer gets up to 5 papers. Middle: Vary the number of papers that each reviewer can review, total number of reviewers is 30. Right: Vary the number of reviewers and vary the number of papers that each reviewer can review according to the number of reviewers we have.

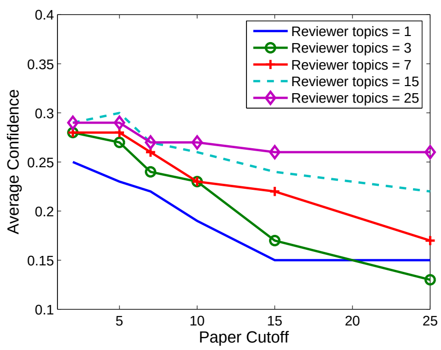

Figure 3: Performance of the ILP algorithm according to Average Confidence measure when papers' topics and reviewers' topics are learned with PLSA model.

4.2 Estimating subtopics with PLSA

When subtopics are unknown, we learn the subtopics for both papers and reviewers using the PLSA model as is done in our previous work[8]. While our ILP algorithm can be directly applied on the probabilistic assignments of subtopics given by PLSA, intuitively, not all the predictions are reliable, especially the low-probability ones. Thus we also experimented with pruning low probability values (i.e., setting low probability elements of P and R to zero). For example, in figure 3, $cutoff_5$ means when we only keep the top 5 probability values out of 25 learned topics and prune the rest. The figure shows the result of the Average Confidence measure. The figure suggests having more topics such as 15 and 25 for reviewers and fewer topics for paper, i.e., $k \geq 4$ and $k \leq 7$ would help improve the performance. This is consistent with the finding reported in [8] (i.e., selecting the modest number of topics often leads to an optimal solution).

5. CONCLUSIONS AND FUTURE WORK

In this paper, we studied the problem of committee review assignment based on multiple subtopics and and proposed a solution based on integer linear programming, which can assign reviewers who would not only have the required expertise to review a paper but also cover all the aspects of a paper in a complementary manner subject to their review quota constraints. Experiment results show that the ILP algorithm is effective.

Due to the lack of resources for evaluation, our evaluation is inevitably preliminary, thus an important future research direction is to further evaluate these algorithms with more data sets ideally applying the algorithms in a real conference. Another interesting extension is to incorporate the *bidding information* into our proposed algorithms.

6. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their useful comments. This work is supported in part by NIH/NLM grant 1 R01 LM009153-01.

7. REFERENCES

[1] C. Basu, H. Hirsh, W. Cohen, and C. Nevill-Manning. Recommending papers by mining the web. *IJCAI*, 1999.

[2] S. Benferhat and J. Lang. Conference paper assignment. *International J. of Intelligent Systems*, 16:1183–1192, 2001.

[3] H. K. Biswas and M. M. Hasan. Using publications and domain knowledge to build research profiles: An application in automatic reviewer assignment. *ICICT 2007*, pages 82–86, 2007.

[4] S. Dumais and J. Nielsen. Automating the assignments of submitted manuscripts to reviewers. *ACM SIGIR*, pages 233–244, 1992.

[5] D. Hartvigsen and J. C. Wei. The conference paper-reviewer assignment problem. *Decision Sciences*, 30:865–876, 1999.

[6] S. Hettich and M. J. Pazzani. Mining for proposal reviewers: Lessons learned at the national science foundation. *ACM SIGKDD*, pages 862–871, 2006.

[7] T. Hofmann. Probabilistic latent semantic indexing. *ACM SIGIR*, pages 50–57, 1999.

[8] M. Karimzadehgan, C. Zhai, and G. Belford. Multi-aspect expertise matching for review assignment. *CIKM*, pages 1113–1122, 2008.

[9] J. J. Merelo-Guervt'os and P. Castillo-Valdivieso. Conference paper assignment using a combined greedy/evolutionary algorithm. *Springer*, 3242:602–611, 2004.

[10] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. *ACM SIGKDD*, pages 500–509, 2007.

[11] C. J. Taylor. On the optimal assignment of conference papers to reviewers. *University of Pennsylvania, Technical Reports*, pages 1–5, 2008.