Recommender Systems for the Conference Paper Assignment Problem

Donald C. Conry

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Applications

Naren Ramakrishnan, Chair
Edward A. Fox
Clifford A. Shaffer

June 4, 2009
Blacksburg, Virginia

Keywords: Collaborative filtering, conference paper assignment.

Recommender Systems for the Conference Paper Assignment Problem

Donald C. Conry

Abstract

Conference paper assignment—the task of assigning paper submissions to reviewers—is a key step in the management and smooth functioning of conferences. We study this problem as an application of recommender systems research. Besides the traditional goal of predicting 'who likes what?', a conference management system must take into account reviewer capacity constraints, adequate numbers of reviews for papers, expertise modeling, conflicts of interest, and an overall distribution of assignments that balances reviewer preferences with conference objectives. Issues of modeling preferences and tastes in reviewing have traditionally been studied separately from the optimization of assignments. In this thesis, we present an integrated study of both aspects. First, due to the sparsity of data (relative to other recommender systems applications), we integrate multiple sources of information to learn reviewer/paper preference models, using methods commonly associated with merging content-based and collaborative filtering in the study of large recommender systems. Second, our models are evaluated not just in terms of prediction accuracy, but also in terms of end-assignment quality, and considering multiple evaluation criteria. Using a linear programming-based assignment optimization formulation, we show how our approach better explores the space of potential assignments to maximize the overall affinities of papers assigned to reviewers. Finally, we demonstrate encouraging results on real reviewer preference data gathered during the IEEE ICDM 2007 conference, a premier international data mining conference. Our research demonstrates that there are significant advantages to applying recommender system concepts to the conference paper assignment problem.

Acknowledgements

The author acknowledges the approval of Prof. Xindong Wu, Steering Committee chair of the IEEE ICDM conference series for the use of preference/bid data collected during the ICDM'07 conference reviewing process, and associated information about papers/reviewers. All datasets were suitably anonymized before the modeling and analysis steps conducted here. The formation of reviewer/paper prediction models was done in collaboration with Dr. Yehuda Koren of Yahoo! Research (Israel). Dr. Koren's insight and previous work was instrumental in the building of these models. Grateful acknowledgement also goes to my faculty advisor, Dr. Naren Ramakrishnan, and my defense committee, Dr. Edward Fox and Dr. Clifford Shaffer; without their knowledge and direction, this thesis would not be possible.

Contents

1 Introduction **1**

1.1 Common terminology 2

1.2 Contributions 3

2 Previous Work **5**

2.1 Bipartite matching 5

2.1.1 Bicliques & closed sets 5

2.2 Constraint-based models 7

2.3 Topic-based models 8

2.4 Network flow models 9

2.5 Small-world networks 10

2.6 Collaborative filtering 11

2.7 Hybrid models 12

2.8 Practical implementations 13

2.9 Limitations of current approaches 14

3 Proposed Approach **15**

3.1 Prediction models 15

3.1.1 Baseline model 16

3.1.2 Latent factor model 17

3.1.3 Subject categories 17

3.1.4 Paper & reviewer similarity metrics 18

3.2 Balancing predictions & preferences 19

4 Evaluation **21**

4.1 Data preparation 21

4.1.1 Preference value scale 22
4.1.2 Mining conflict of interest data 22
4.2 Assignment performance 24
4.3 Prediction quality 25
4.4 Analysis of predicted values 27

5 Conclusion & Future Work **32**

Bibliography **33**

List of Figures

1.1 Common data matrices for conferences. 2

2.1 A simple bipartite matching problem, as discussed in [35]. 6

2.2 Connection between bicliques and closed sets. The numbers in the lattice are closed sets of reviewer IDs from the matrix of assignments. Numbers in parenthesis are frequencies of closed sets. 7

2.3 Illustration of the 'small-world' property from [20]. 11

4.1 Performance comparison of various new models against unmodified assignments. . 25

4.2 Topical relevance of assignments made with our approach versus Taylor's original formulation. 26

4.3 Evaluating the assignments made by the unmodified Taylor algorithm and the new preference models w.r.t. reviewers' four categories of preferences, using a 70-30 test-training set split, averaged across 20 iterations. Mean assignments per iteration are indicated above each bar. 29

4.4 Evaluating the assignments made by the unmodified Taylor algorithm and the new preference models, using a 60-40 test-training set split, averaged across 20 iterations. Mean assignments per iteration are indicated above each bar. 29

4.5 Evaluating the assignments made by the unmodified Taylor algorithm and the new preference models, using a 50-50 test-training set split, averaged across 20 iterations. Mean assignments per iteration are indicated above each bar. 30

4.6 Comparison of one of our chosen models to a hybrid model which modifies only unknown preferences (labeled 'unk'). 30

4.7 Prediction range and standard deviation per reviewer. While the range is often relatively large, predicted values typically appear to cluster close together, indicating outliers. 31

List of Tables

4.1 Percentage of assignments made from the ‘unknown’ test set members, broken down by actual known preference values. Our methods show improvement over the unmodified Taylor LP, assigning a much higher percentage of ‘preferred’ papers. 27

Chapter 1

Introduction

Modern conferences, especially in areas such as data mining/machine learning (KDD; ICDM; ICML; NIPS) and databases/web (VLDB; SIGMOD; WWW), are beset with excessive paper submissions. Assigning these papers to appropriate reviewers in the program committee (which can constitute a few hundred members) is a daunting task and hence motivates the use of recommender systems.

Recommender systems have traditionally been studied in e-commerce domains such as books, food, music, and movies. However, these systems are applicable in practically any domain where the goal is to model user preferences and achieve objectives of bringing people and artifacts together. Here we study the use of recommender systems for the conference paper assignment problem (CPAP), which involves systematically assigning reviewers to papers for the peer-review system typical to many academic conferences. Just as e-commerce product recommenders model partial data about people and the products they like, the goal in CPAP is to model data about people and the papers they would like to review. There are also more issues to be considered for conference management than in traditional recommender systems applications.

Besides the traditional goal of predicting 'who likes what?', a conference management system must take into account aspects such as reviewer capacity constraints, adequate numbers of reviews for papers, expertise modeling, conflicts of interest, and an overall distribution of assignments that balances reviewer preferences with conference objectives. These diverse considerations can lead to a number of problems with assignments; among these, issues of modeling preferences, expertise, and tastes in reviewing have traditionally been studied separately from the problem of improving reviewer to paper assignments. The former has been the subject of much academic research (e.g., see Section 2.3) while the latter is emphasized by software, such as EasyChair [6], CyberChair [41], and Microsoft's CMT [1], which aim to automate the management of the conference reviewing process.

CPAP is a specific example of the general assignment problem in discrete mathematics, where the solution is given by a matching of a weighted bipartite graph consisting of agents (reviewers) and tasks (papers). We first present some common terminology and concepts.

1.1 Common terminology

Conference management research styles itself towards the terms and resources used by an individual conference or group of conferences featured in that research. However, there are a number of concepts common to most or all existing approaches. It is useful to develop and formalize a common terminology, even though not all of the concepts discussed below are included in every approach.

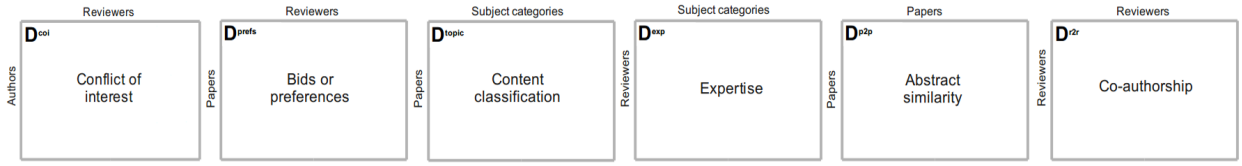

Figure 1.1: Common data matrices for conferences.

A reasonable assumption is that a reviewer should not be assigned to review a paper for which he or she is the author. Many conferences also extend this restriction to include close associates (*affiliates*); for example, a conference may stipulate that reviewers may not be assigned papers authored by individuals from the same academic institution as the reviewer. Other conferences restrict reviewing of papers authored by individuals that collaborated on a different paper with the reviewer in the past. Some conferences ask reviewers to manually indicate when a conflict of interest exists between themselves and one or more authors. This type of restriction is considered conflict of interest data, and is represented by matrix $\mathbf{D}^{coi}$ in Fig. 1.1. The reflexive conflict involving an author who is also a reviewer for a given conference is sometimes considered as self-evident; it also can be represented in matrix $\mathbf{D}^{coi}$ if this type of conflict is considered.

Another type of data commonly gathered is the preference of each reviewer for one or more papers to be assigned. For example, each reviewer might be asked to rank as many papers as possible across a range of possibilities, such as 'Would like to review', 'OK to review', 'Prefer not to review', and 'Cannot review'. Some CMS software simply allows reviewers to 'bid' for papers, corresponding to only the 'Would like to review' option from the previous example. This preferences or bidding data occasionally is used in place of (or in parallel with) conflict of interest data, by reserving the lowest preference option in matrix 'Cannot review' for just such a conflict, and expecting the reviewer to choose that option if a conflict exists. Other times this data is kept completely separate, and is represented as matrix $\mathbf{D}^{prefs}$ in Fig. 1.1.

Some conferences also consider topical data for sub-topics within the conference's target discipline, ranking either papers or reviewers (or both) for each category or topic. These rankings can be binary to indicate inclusion or exclusion for a certain topic, integral to differentiate between stepped levels of familiarity with a topic, or continuously-valued. In any of these cases, the matrix of topics to reviewers contains a score for each reviewer/topic combination (matrix $\mathbf{D}^{exp}$ in Fig. 1.1), and papers can similarly be ranked in each topic (matrix $\mathbf{D}^{topic}$). A reviewer/topic matrix is one way to model *expertise*, while paper/topic matrices show *content classification* for each paper.

Keywords are another data source used for CPAP; keyword relevance for each reviewer and

paper can either be assigned manually by survey, or mined automatically from paper content such as abstracts (or from the web). Topical relevance is very similar to keyword relevance; the above model is also applicable to storing keyword data. Conferences sometimes use topical and keyword data to compute measures of similarity between the individual papers (matrix $\mathbf{D}^{p2p}$) or between individual reviewers (matrix $\mathbf{D}^{r2r}$); this is comparable to data frequently used in recommender systems.

Merging the conflict of interest or topical data with the $\mathbf{D}^{prefs}$ matrix would require a more sophisticated set of transformations. However, the underlying idea of combining disparate sources of information into a single set of values (or scores) is intriguing. These merged values could then be used as input data for any of the algorithms described in Chapter 2. A more specific approach to combining the different input values will be discussed afterwards. Note that the weights given to the original values from each data source would be a configurable parameter affecting the final solution.

1.2 Contributions

We investigate the conference paper assignment problem through the lens of recommender systems research. There are three key differences between CPAP and traditional recommender systems; first, in a traditional recommender, meeting the needs of one user does not affect the satisfaction of other users. In CPAP, multiple users (reviewers) are bidding to review the same papers, meaning that one user's recommendations (assignments) may negatively affect the satisfaction levels of other users. Therefore, the design of reviewer preference models must be posed and studied in an overall optimization framework.

Second, in a conventional recommender, the goal is often to recommend *new* entities that are likely to be of interest, whereas in CPAP, the goal is to ensure that reviewers are predominantly assigned their (most) preferred papers. Nevertheless, preference modeling is still crucial because it gives the assignment algorithm some degree of latitude in satisfying multiple users.

Finally, recommender systems are often used with sparse data, but the amount of 'signal' available to model preferences in the CPAP domain is exceedingly small. Thus, we must integrate multiple sources of information to build strong preference models.

In this thesis, we present the first integrated study of both modeling reviewing preferences and optimizing assignments for conference management. Our key contributions can be summarized as:

1. Due to the paucity of data per reviewer or per paper (relative to other recommender systems applications) we show how we can integrate information about publication subject categories, contents of paper abstracts, and co-authorship information to learn improved reviewer/paper preference models.

2. We evaluate our models not just in terms of prediction accuracy but also in terms of the end-assignment quality. Using a linear programming-based assignment optimization for-

mulation, we show how our approach better explores the space of potential assignments to maximize the overall affinities of papers assigned to reviewers.

3. We demonstrate the effectiveness of our approach on actual reviewing preference data in the context of a real life conference, namely the IEEE ICDM'07 conference [34]. Due to privacy concerns, real life conference data is difficult to obtain; although we demonstrate our approach's effectiveness on only one dataset, we hope that this research would lead to additional available datasets, and that our successes will translate to these other datasets as well.

The rest of the thesis is organized as follows. Chapter 2 is a review of previous related work, spanning different kinds of conference paper review assignments, collaborative filtering, and other topics related to conference assignment. It also contains a survey of existing conference management software. Chapter 3 details our framework for a new approach, with the specifics of models used to predict missing preferences and assign papers to reviewers. Chapter 4 includes an evaluation of both predictive accuracy and assignment performance; existing criteria are considered and new criteria are developed. Chapter 5 provides a summary of the salient points of the thesis, and a list of topics for further research.

Chapter 2

Previous Work

Conference management systems (CMS) are software applications designed to help with the organization and administration of conferences. There are a number of issues involved while organizing a conference; for the purposes of this thesis, we mainly consider features that deal with assigning papers to reviewers via preferences and expertise modeling. The result from any solution to CPAP should be a set of assignments of reviewers to papers for the conference. Perhaps the simplest solution to CPAP would be to spread the papers across all available reviewers randomly. This chapter presents an overview of approaches to this problem that try to improve on the trivial random assignment. Like the input data, these assignments can be represented in matrix form.

2.1 Bipartite matching

A popular approach is to consider CPAP as a bipartite matching problem. This approach models reviewers and papers as vertices of a graph connected via weighted edges, where the weights represent the preference of a reviewer for a paper. The objective is to connect the vertices of one type to the other, while maximizing the sum of the edge weights. The classical solution is given by the Hungarian Algorithm described by Kuhn [24]; it provides a solution for the simplest cases of this family of problems (applicable when the number of reviewers equals the number of papers; see Fig. 2.1). Various refinements have been made to this algorithm over the years, such as one by Hopcroft and Karp [19]. This approach is taken by a number of contemporary assignment systems, including GRAPE [12] and the MyReview system [2]. Maximizing the bipartite edge weights corresponds nicely to maximizing the various reviewer to paper relevance scores used in these systems.

2.1.1 Bicliques & closed sets

A biclique or complete bipartite graph consists of two sets of vertices where every vertex of the first set is connected to every vertex of the second set. In CPAP, the matrix of review assignments **R**

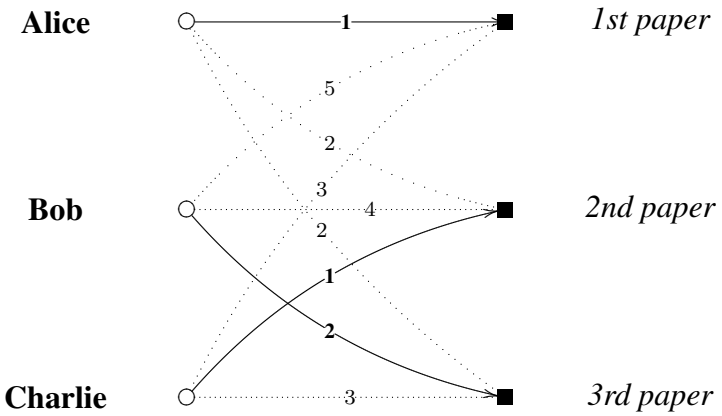

Figure 2.1: A simple bipartite matching problem, as discussed in [35].

serves as the *adjacency matrix* of a bipartite graph **G**, where **G** contains edges connecting reviewers to assigned papers. A group of reviewers that share review assignments for the same paper or papers therefore forms a biclique in graph **G**.

Li et al. [26] pointed out that the adjacency matrix of such a graph can be viewed as a transactional database, and a one-to-one correspondence exists between maximal closed itemsets (or patterns) and maximal biclique subgraphs (see Fig. 2.2). Therefore, bicliques can be discovered by finding the frequent closed itemsets in the adjacency matrix, a task for which many data mining algorithms exist. Li et al. [27] further develop this idea, and introduce techniques to remove wasted work done involving duplicate sets. This close correspondence allows algorithms such as CHARM [46] to identify bicliques in graph **G**. Additionally, the lattice representation of closed sets from the variant algorithm CHARM-L maps the structure of these bicliques in a graph.

Bicliques in a graph tend to increase the *clustering coefficient* (also called *network transitivity*), while decreasing the *average (shortest) path length* between vertices. As defined by Newman [31], the clustering coefficient C of a network measures the degree that neighbors of a given vertex are also neighbors of each other. A simple formula for calculating the clustering coefficient is also given in terms of connected triples (a single node connected to two others), and is easy to calculate manually:

$$C = \frac{3 \times \text{number of triangles in the network}}{\text{number of connected triples of vertices}} \tag{2.1}$$

A more complex version that is usually computed by algorithm, and is often used for statistical and data analysis purposes, is given by Watts and Strogatz [42]:

$$C = \frac{\text{number of 'triangles' connected to vertex } i}{\text{number of 'triples' centered on vertex } i} \tag{2.2}$$

These quantities are not exactly equivalent, but both tend to be higher for tightly clustered (regular) graphs and lower for unclustered (random) graphs [31].

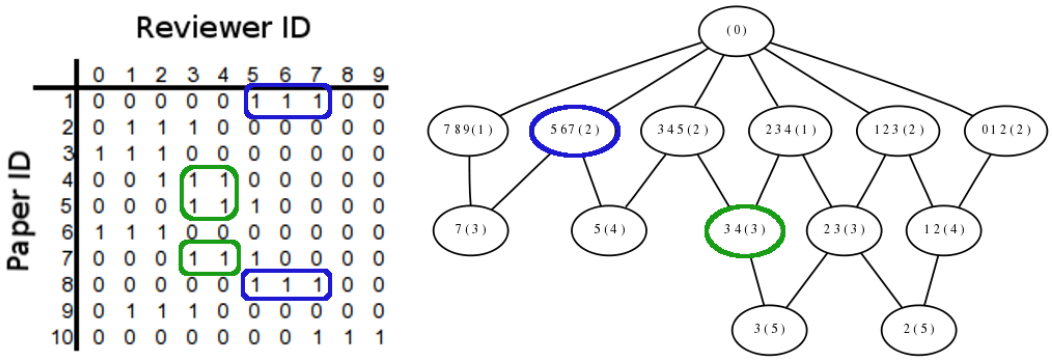

Figure 2.2: Connection between bicliques and closed sets. The numbers in the lattice are closed sets of reviewer IDs from the matrix of assignments. Numbers in parenthesis are frequencies of closed sets.

2.2 Constraint-based models

For practical reasons, restricting the number of reviews per reviewer and per paper to a specific interval is a natural step for assigning reviewers. Assigning too many reviews to a particular reviewer may result in insufficient time for that reviewer to complete his reviews; assigning too few reviews would result in insufficient opinions with which to fairly judge a given paper. Obviously, the specifics of these intervals depends on the individual conference, but reasonable constraints are generally chosen by the conference chair or agreed upon by the program committee, for example. As noted above, reviews can be assigned randomly based purely upon these per-reviewer and per-paper constraint values. The problem with this type of solution is a lack of consideration for whether or not an assigned reviewer has the topical knowledge or expertise to accurately judge the paper.

A natural extension would be to add a constraint to represent the expertise factor. In a paper by Taylor [40], an *affinity* constraint is created to fill this role. The set of assignments then can be quantitatively judged based on the global affinity attained. This affinity constraint can be defined in a number of different ways; Taylor documents the use of area chairs to rank authors based on personal knowledge of each reviewer's expertise. Another possibility would be to ask the reviewers themselves to indicate their preferences for reviewing (or not reviewing) each paper. These affinity values can be compiled into a matrix of all reviewer/paper combinations, and used to solve the original constraint-satisfaction problem. The affinity values effectively rank reviewers for each paper, similar to the effect of the other models above.

Taylor's affinity matrix can be represented by any of the input criteria defined in Chapter 1, or any reasonable combination of them; for example, $\mathbf{D}^{prefs}$ would be a natural candidate for such an affinity matrix. Denoting the affinity matrix as simply $\mathbf{D}$ in equation (2.3) will indicate the general use of one or more of the input criteria, as desired for any given conference. The specific choice of affinity measure does not affect Taylor's optimization formulation, expressed in terms of $\mathbf{D}$ and the assignments matrix $\mathbf{R}$, as:

$$\begin{aligned} \underset{\mathbf{R}}{\operatorname{argmax}} \quad & \text{trace}\left(\mathbf{D}^T\mathbf{R}\right) = \sum_i \sum_j \mathbf{D}_{ij}\mathbf{R}_{ij}, \\ & \text{where} \qquad \mathbf{R}_{ij} \in [0,1] \qquad \forall i,j, \\ & \text{and} \quad \sum_j \mathbf{R}_{ij} \leq c_r, \qquad \forall i, \\ & \text{and} \quad \sum_i \mathbf{R}_{ij} \leq c_p, \qquad \forall j. \end{aligned} \tag{2.3}$$

where c_p represents the desired number of reviews per paper, and c_r is the desired maximum reviews per reviewer. The third and fourth lines in equation (2.3) represent the constraints on the number of assignments for individual papers and to individual reviewers, respectively. Then the expression *trace* $\left(\mathbf{D}^T\mathbf{R}\right)$ represents the global sum of affinity or happiness of all reviewers across all assigned papers. In particular, by using the (binary) assignments matrix $\mathbf{R}$ as a factor, only the affinities from $\mathbf{D}$ for reviewer/paper combinations that exist in the final assignments $\mathbf{R}$ are counted in the sum.

This integer programming problem (2.3) is reformulated into an easier to manage linear programming problem by a series of steps, using the node-edge adjacency matrix, where every row corresponds to a node in $\mathbf{R}$, and every column represents an edge. This linear reformulation is a bit more complicated, but retains the same meaning as the integer programming problem above, and can be solved with relative quickness using existing methods (such as the *linprog()* solver in *MATLAB*).

In particular, as Taylor shows in [40], because the reformulated constraint matrix is *totally unimodular*, there exists at least one globally optimal solution (assignment set) with integral (and due to the constraints, Boolean) coefficients. Taylor ensures a unique solution by slightly perturbing the values (randomly), in order to impose a rank on equivalently valued nodes. While this does ensure a unique optimal solution, it seems that a more deterministic method of imposing this rank on like-valued nodes may prove more useful as a solution to the original problem.

2.3 Topic-based models

One view of conference review assignments is that papers should be assigned to reviewers with a certain degree of expertise in the specific field or topic of the paper. This view leads to topic-based approaches that use additional information to improve paper assignments; namely, the main topic or topics of each paper, as well as (in some cases) the area or areas of expertise for each reviewer. By using this information, reviewer assignments can be tailored to suit each paper's topic, ensuring a degree of expertise is present. The resultant ranking of each reviewer based on topical knowledge with respect to a given paper has been called *expert-finding* and *expertise modeling* [30].

One problem involved with this approach is identifying which topics are covered in papers. Early efforts in this area focused mainly on paper abstracts, and topical expertise was determined through common information retrieval methods involving keywords. For example, Dumais and Nelson [13]

match papers to reviewers using Latent Semantic Indexing trained on reviewer-supplied abstracts. In Basu et al. [8], abstracts from papers written by potential reviewers are extracted from the web via search, and a vector space model is constructed for matching. Yarowsky and Florian [45] extended this idea by using a similar vector space model with a naive Bayes classifier run against work previously published by each reviewer. More recently, Wei and Croft [43] describe a topic-based model using a language model with Dirichlet smoothing.

An excellent example of topic-based models is the Author Persona Topic (APT) model by Mimno and McCallum [30]. The APT model contains a number of features designed to better capture the reality of the relationship between conference reviewers and papers. The basic idea is that an author may study and write about several distinct topics; by clustering papers from each of these topics into a separate persona for an author, the author's ranking for a given topic need not be diluted by his or her writings on a different topic. Again, the system was trained using abstracts of the potential reviewers (in effect, mining data equivalent to $\mathbf{D}^{exp}$), and the APT model was found to perform better than other language-based and topic-based models.

2.4 Network flow models

Another approach to CPAP uses reasoning from the much more general *minimal cost network flow problems* studied in dynamics and operations research. Many such related problems (known collectively as extended generalized assignment problems [10] or GAP) of assigning a limited number of resources to certain tasks exist in diverse fields. They include personnel assignment or scheduling, optimal usage of inventory, vehicle/airline scheduling, and many others (e.g., see [17]). In the network flow diagram of this general assignment problem, resources (in our case, reviewers) are represented by source nodes with a certain supply (number of reviews allowed per reviewer), while tasks (each paper to be reviewed) are sink nodes with a demand (the number of times each paper must be reviewed). While an optimal solution to this problem can be found, this formulation does not consider the 'expertise' factor as described in the previous model.

Hartvigsen et al. [17] describe an extension of the basic network minimum cost flow (MCF) model that is constructed specifically to solve the problem of assigning reviewers to conference papers. A ranking of potential reviewers for each paper is constructed, using the results of a survey where each reviewer is allowed to classify his or her expertise into categories (essentially the data in $\mathbf{D}^{exp}$); the papers are similarly classified (corresponding to $\mathbf{D}^{topic}$). However, the key idea is that each reviewer is given a maximum number (m_p) of 'points' to use in assigning expert topics to themselves. These m_p points can be spread one by one over many categories (up to m_p different topics), or combined to give greater weight to expertise in a few topics. Similarly, each author is given m_p points to classify the main topics of his or her submitted paper. These classification vectors for reviewers and authors can be modeled by our previously defined matrices $\mathbf{D}^{exp}$ and $\mathbf{D}^{topic}$, respectively. Note that in this case, for all topics i, the aggregate measure of expertise for each reviewer can be written as $\sum_i \mathbf{D}^{exp}_{ij}$ where j stands for reviewer, and aggregate topical relevance for each paper as $\sum_i \mathbf{D}^{topic}_{ik}$ where k stands for paper. Then:

$$m_p = \sum_i \mathbf{D}_{ij}^{exp} = \sum_i \mathbf{D}_{ik}^{topic} \qquad \text{for all reviewers } j, \text{papers } k. \tag{2.4}$$

A weighted measure of similarity between papers and reviewers is then calculated. By assigning these weights to the associated edges in the network, a measure of expertise is imposed onto a standard network flow problem [15]. This results in a linear optimization problem to maximize the flow between source nodes (reviewers) each with a certain supply of reviews, and sink nodes (papers) with a certain demand for reviews. The problem is then reformulated to add additional constraints on the solution. First, a requirement is made that every paper is assigned at least one reviewer whose level of expertise ranking to that paper meets a certain global threshold. Second, that global threshold is maximized with respect to the original constraints, thus finding the highest global threshold that allows for a feasible solution to the original problem.

2.5 Small-world networks

A number of real-world network topologies contain some properties of both regular and random graphs [31]. In general, regular graphs have a high degree of clustering, meaning that neighbors of a node are usually also connected, resulting in a very localized structure as shown on the left in Fig. 2.3. This localized structure also results in a comparatively high average path length, since nodes on opposing sides of the 'ring' in the figure will have especially long distances between them. In contrast, random graphs generally have low average path lengths; as depicted on the right in Fig. 2.3, 'shortcuts' across the middle of the graph are fairly common, serving as bridges that facilitate shorter path lengths between arbitrary nodes.

Watts and Strogatz [42] define a third type of network graph called a 'small-world' network, as shown in the center of Fig. 2.3. These small-world networks retain the high clustering coefficient (regardless of which definition is used) of regular graphs, while also having short average path lengths between nodes, as in random graphs. The key realization (generally attributed to Milgram [29]) is that only a few 'shortcuts' are needed to induce short average path lengths in a graph. Once that threshold of shortcuts is reached, and average path length is not reduced much by additional shortcuts. In fact, the network structure can remain highly clustered while still benefiting from short paths between arbitrary nodes.

The idea can be illustrated by starting from a regular graph involving very high local clustering and very long paths between non-local endpoints. The graph structure can be altered by visiting each node in the graph, and randomly 'rewiring' (disconnecting and reconnecting to a random node in the graph) with probability p. If $p = 0$, then no edges are rewired and the graph remains regular, as shown on the left in Fig. 2.3. Alternatively, if $p = 1$, then every edge is rewired randomly, resulting in a random graph such as the one depicted on the right side of the figure. Values of p between 0 and 1 represent varying degrees of rewired edges. There are many simulations of such rewirings in [42], and both the average path length ($L(p)$) and clustering coefficients ($C(p)$) are expressed as functions of the original rewiring probability p. At a certain value of p, $L(p)$ decreases swiftly even while $C(p)$ remains largely unchanged, resulting in the high clustering/low average path length graph structure discussed above.

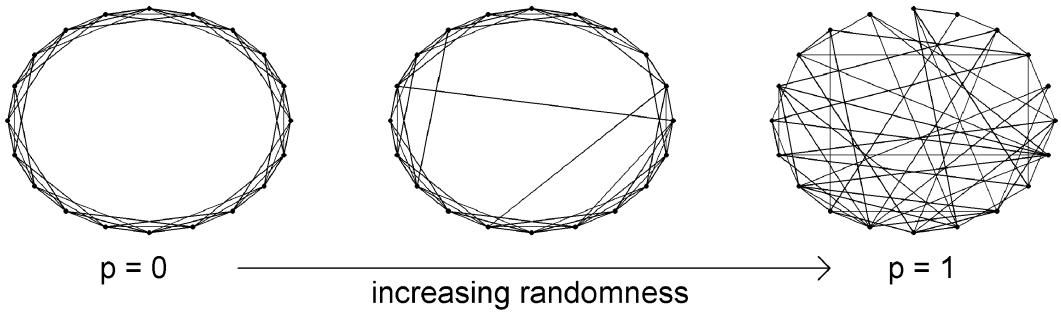

Figure 2.3: Illustration of the 'small-world' property from [20].

This type of small-world network structure seems to be ideal for conference assignment graphs. Academic conferences usually focus on some number of *related* topics, so there is quite a bit of overlap in both expertise and preferences among reviewers for these topics. In a small-world assignment graph, the reviewers and papers can be clustered around topics, and still retain a degree of overlap among those topics. Humphries and Gurney [20] introduce a quantitative measure of 'small-worldness' (SW) for graphs. Depending upon the goals of the specific conference, this SW measure can be used to tune an assignment graph, inducing either more or fewer 'random' edges. The 'rewiring' procedure, instead of being done randomly as in [42], can be done by considering the existing assignments and deciding (based on factors such as those discussed in the beginning of this section) which nodes are 'bad' and can be rewired. An additional step could even be added to the assignment process, allowing users to indicate which assignments they dislike, then attempting to rewire the indicated assignments.

2.6 Collaborative filtering

Several of the approaches to CPAP detailed above make use of preference or bid data from reviewers. A weakness of this approach is the typical sparsity of such information; most reviewers return preferences only for a small percentage of papers. Rigaux [35] suggests the use of collaborative filtering techniques to grow the preference data.

Early examples of collaborative filtering (CF) are Tapestry [14] and the GroupLens project [21]. The basic assumption (as applied to the $\mathbf{D}^{prefs}$ matrix defined above) is that reviewers who bid similarly on a number of the same papers will likely have similar preferences of other papers. For example, if reviewer A's bids closely match those of reviewer B for papers 1, 2, and 3, and reviewer A has bid positively on paper 4, there is some indication that reviewer B would likely also bid positively on paper 4, based on past agreement with A.

Naturally, some degree of similarity between reviewers is needed; Rigaux uses the popular Pearson correlation coefficient. If $\bar{r}_i$ is the average bid or preference expressed by any user i in $\mathbf{D}^{prefs}$, then the Pearson correlation $\mathbf{corr}_{kl}$ between reviewers k and l is given by:

$$\text{corr}_{kl} = \frac{\sum_j \left(\mathbf{D}^{prefs}_{kj} - \bar{r}_k\right)\left(\mathbf{D}^{prefs}_{lj} - \bar{r}_l\right)}{\sqrt{\sum_j \left(\mathbf{D}^{prefs}_{kj} - \bar{r}_k\right)^2 \sum_j \left(\mathbf{D}^{prefs}_{lj} - \bar{r}_l\right)^2}} \tag{2.5}$$

Thus equipped with a measure of the correlation of bids for each pair of users, missing bids in $\mathbf{D}^{prefs}$ can be "filled in" by altering the default (missing) values in a direction suggested by other reviewers that have bid on an item, and that are highly correlated with a reviewer that has not bid on that item. Rigaux further justifies these alterations in $\mathbf{D}^{prefs}$ by attempting to measure the *significance* of each alteration, based upon a standard number K of correlations for which a change should be deemed relevant.

The assumption is that the correlation between reviewers with fewer than K paper bids in common is too small to be considered. Indeed, with smaller conferences the lack of overlap between bids from reviewers becomes a problem. Rigaux recommends asking users to bid on all or most of the papers in a given topic, instead of a few bids over the entire set of papers. This encourages a higher degree of bid overlap, allowing more of the calculated correlations to be considered relevant. A second recommendation is that reviewers with a low degree of correlation to others be re-polled for additional bids, in order to increase the relevance of their preferences in regards to the other reviewers.

2.7 Hybrid models

Basu et al. [8] use the relational WHIRL system to multiply similarity scores from disparate data sources into a single aggregate similarity score. WHIRL uses methods from the field of information retrieval; it constructs a vector space (TF-IDF) similarity model to compare papers or reviewers. The system is then queried for the top **n** similar items for a given paper or reviewer. This system is very similar to search in IR, and thus results can be ranked by precision. The authors do not attempt to satisfy per-paper or per-reviewer constraints, and the contributions of different sources are equivalent to each other.

Popescul et al. [33] present a way to combine content-based and collaborative recommendations using a three-way aspect model. The authors use conditional probabilities and expectation maximization to discover optimal weights for the different data sources. Especially in conditions where primary data is sparse, secondary data improves the quality of recommendations by simulating increased data density. This is similar in direction to the approach presented below, but uses a probabilistic model instead.

The GRAPE system [12] prefers topical information from $\mathbf{D}^{exp}$ and $\mathbf{D}^{topic}$ over the reviewer bids or preferences as gathered in $\mathbf{D}^{prefs}$; however, preferences *are* used as a secondary means of assignment (i.e. are given less weight in the assignment process). The reasoning behind this decision is the assumption that the topical data more accurately predicts the degree of expertise present for a reviewer/paper match. Since the distribution of reviewers and papers over topics is unpredictable (sometimes leaving too many or too few reviewers for a given cluster of papers),

the preference information is used for tuning or smoothing out the wrinkles in the topic-based assignments.

The implementation of GRAPE makes use of a degree of *confidence* between reviewers and papers. This degree of confidence is defined as the number of topics in common between the reviewer and paper. More formally, given m total topics under consideration, let $\vec{r}_j = (\mathbf{D}^{exp}_{1j} \ldots \mathbf{D}^{exp}_{mj})$ represent the vector of topical expertise values for reviewer j, and let $\vec{p}_k = (\mathbf{D}^{topics}_{1k} \ldots \mathbf{D}^{topics}_{mk})$ be the vector of topical content values for paper k. Then the degree of confidence $\mathbf{conf}_{jk}$ between reviewer j and paper k is defined as the total number of topics in common between these vectors:

$$\text{conf}_{jk} = \vec{r}_j \cdot \vec{p}_k \tag{2.6}$$

This quantity is combined with data equivalent to the preferences matrix $\mathbf{D}^{prefs}$ to produce a *gratification degree* g_{r_j} for each reviewer j. Similarly for papers, the confidence degree of each reviewer/paper match is combined with the *coverage degree* after the assignments for each paper, defined as the percentage of all topics for which one of the assigned reviewers is an expert. Both gratification and coverage degrees are maximized by GRAPE during the review assignments. These two quantifiable heuristics are not evident from CPAP directly, but they are somewhat indirectly related to objectives common to other CPAP approaches, such as the Pearson correlation-inspired heuristics used by Rigaux [35].

The SimFusion algorithm and Unified Relationship Matrix presented by Xi et al. [44] involves augmenting a sparse primary data matrix by integrating multiple secondary data matrices. Though this work is not specifically aimed at conference data, the fundamental goals are similar in flavor to the methods presented in the next chapter. Applying the SimFusion algorithm to conference data may prove interesting as a topic for future research.

2.8 Practical implementations

This section provides a representative (but not exhaustive) survey of conference management systems. The software has been collected from various sources, including word-of-mouth, web search, and the bibliographical references from the literature (especially [17] and [20]). Two of the software packages (START and Microsoft's CMT) are commercial in origin; the rest are mostly of academic origins, and freely available for use by academic conferences. A summary of the features included with various older CMS software packages is available from the ACM website [39]; the list below focuses on software with automatic paper review assignment features.

Software	Auto assign?	Considers bids/prefs	Considers topics	License & comments
BYU PRS	✓	✓	✓	Free; documented at [4]
ConfMan		✓		Free; documented at [5, 16]
Continue 2.0			✓	Free; documented at [23]
CyberChair	✓	✓	✓	Free; based on Nierstrasz [32]
EasyChair	✓	✓		Free; documented at [6]
GRAPE	✓	✓	✓	Based on Di Mauro et al. [12]
Microsoft CMT	✓	✓		Free for non-commercial use [1]
MyReview	✓	✓		Free; based on Rigaux [35]
OpenConf	✓		✓	Free; documented at [7]
START	✓	✓		Commercial from [3]

2.9 Limitations of current approaches

Many of the algorithms mentioned above focus on a single input data source to generate assignments. For example, Taylor's linear program [40] and Rigaux's collaborative filtering approach [35] only consider reviewer bids ($\mathbf{D}^{prefs}$). In Taylor's approach, potential assignments with equivalent 'affinities' or bid values are chosen in random order. Rigaux provides a ranking of these potential assignments with equivalent affinities, using Pearson correlation between the preference vectors of similar reviewers. This affects the order in which reviews with equal affinities are assigned, but still only considers one source of input data.

The GRAPE [12] and CyberChair [41] systems are notable for considering both reviewer bids and topical expertise, using two-phase assignment processes. This adds the benefit of multiple input sources, but in each case the secondary source is only used when the first fails to provide any preference. Instead, the secondary source of data could be used to influence assignments in a way similar to Rigaux's CF approach. This allows for single-phase assignments, where elements of both data sources can influence each assignment in the resulting matrix $\mathbf{R}$. Both Basu [8] and Popescul [33] present methods that do this for similarity score data, but these methods do not consider non-continuous data values (e.g., binary conflict of interest and affiliation data); also, in both cases the transformation into combined data is set and well-defined.

We will present models that consider an arbitrary number of input sources, in a way that is flexible enough to meet the needs of a wide range of conferences.

Chapter 3

Proposed Approach

This thesis proposes a way to consider an arbitrary number of data sources for the purposes of assigning reviews, by combining data from these sources into a single aggregate matrix **D**. Using techniques from collaborative filtering and recommender systems, the combined matrix can be viewed as a prediction problem. In particular, models developed by Koren [22] are equally useful here in merging relevance scores from multiple input sources ($\mathbf{D}^{topic}$, $\mathbf{D}^{exp}$, $\mathbf{D}^{coi}$, etc.) into an integrated predictions matrix **P**. These predictions augment a sparse primary dataset ($\mathbf{D}^{prefs}$) to produce the final input matrix (**D**). This matrix can be used as the input to many of the existing linear programming or bipartite matching algorithms.

In the following section, we gradually expand the prediction model, by introducing into it a growing set of features. Running this model against the full set of potential assignments results in a matrix **P** of predicted bids. Results of the prediction model will be embedded into an assignment formulation, namely the Taylor linear programming algorithm. Although we use the same assignment formulation as Taylor, Chapter 4 will present new ways to evaluate the assignments.

Section 3.1 describes models developed for this thesis, but for any given conference these models can vary depending on the data available and the goals intended. The specifics of the data used for this thesis are presented in Section 4.1.

3.1 Prediction models

Because of the sparsity of preferences with respect to the number of possible review assignments, learning these models using a single test-train split of known preferences could be adversely affected by irregularities in the sets. Instead, we use cross-validation; we calculate 100 different randomly seeded splits of the known preferences. 90% of these known values are placed in $|TrainingSet|$; these values are used in learning the models, via a least-squares optimization of the model parameters. The remaining 10% are placed into $|TestSet|$; these values will be used to evaluate the predictive accuracy of the learned models. The quality of the results on a specific test set is measured by their root mean squared error (RMSE) when compared to the actual preference values. The overall accuracy of the model is taken as the mean RMSE over 100 randomly

generated test sets.

We hasten to add that we do not advocate the myopic view of RMSE [28] as the primary criterion for recommender systems evaluation. We use it in this chapter primarily due to its convenience for constructing direct optimizers. In the next chapter we will evaluate performance according to criteria more natural to the paper assignment problem. We also note that small improvements in overall RMSE will typically translate into substantial improvements in bottom-line performance for predicting reviewer/paper preferences.

The following models proved (through extensive experimentation) to be useful in forming a prediction model for ICDM'07. Different data sources may be more appropriate for any given conference, but these also fit the generic method described above.

3.1.1 Baseline model

The simplest model would be to select a single global (mean) bid μ to minimize the least squares difference between predicted and actual preference values. This results in a series of equations for each reviewer i. In the case of conference assignments, we can include separate bias vectors for reviewer and paper (denoted $\vec{\tau}$ and $\vec{\rho}$, respectively). Much of the variability in the data is explained by global effects, which can be reviewer- or paper-specific. Bell and Koren [9] use a similar technique (referred to as "removing global effects") for collaborative filtering. These vectors provide additional constants per reviewer and per paper, in addition to the global bias μ.

$$\mathbf{P}_{ij} = \mu + \tau_i + \rho_j \tag{3.1}$$

We learn optimal values for τ_i and ρ_j by minimizing the associated squared error function f for all known preferences $\mathbf{D}_{ij}^{prefs}$ in the training set:

$$f(\tau_i, \rho_j) = \min_{\tau,\rho} \sum_{i,j} (\mathbf{D}_{ij}^{prefs} - \mu - \tau_i - \rho_j)^2 + \lambda_1 \tau_i^2 + \lambda_2 \rho_j^2 \qquad \forall i, j \in |TrainingSet| \tag{3.2}$$

The regularizing term, i.e., $\lambda_1 \tau_i^2 + \lambda_2 \rho_j^2$ prevents overfitting by penalizing the magnitudes of the parameters. We set the values of the constants λ_1 and λ_2 by cross validation. Learning is done by stochastic gradient descent (alternatively, any least squares solver could be used here). The gradient of equation (3.2) is given by the vector of first order partial derivatives with respect to τ and ρ:

$$\begin{aligned} \nabla f(\tau_i, \rho_j) = \big[& (2\lambda_1 + 2)\,\tau_i + 2\mu + 2\rho_j - 2\mathbf{D}_{ij}^{prefs}, \\ & (2\lambda_2 + 2)\,\rho_j + 2\mu + 2\tau_j - 2\mathbf{D}_{ij}^{prefs} \big] \qquad \forall i, j \in |TrainingSet| \end{aligned} \tag{3.3}$$

The gradient points locally in the direction of the steepest ascent of f, so we choose a small step size η in the direction of $-\nabla f$ (the direction of steepest descent). Given an initial guess of the values, we can iterate and find values for τ and ρ that are successively closer to the locally optimal values.

The resulting average test RMSE is **0.6286**. A separate analysis of each of the two biases shows reviewer effect ($\mu + \tau$, with RMSE **0.6336**) to be much more significant than paper bias ($\mu + \rho$, RMSE **1.2943**) in reducing the error. This indicates a tendency of reviewers to concentrate all bids near their mean bid values, which is supported by examination of the data.

3.1.2 Latent factor model

Latent factor models comprise a common approach to collaborative filtering with the goal to uncover latent features that explain observed values; examples include pLSA [18], neural networks [37], and Latent Dirichlet Allocation [11]. We will focus on models that are induced by factorization of the reviewer/paper preferences matrix, resulting in vectors $\vec{p}_i$, which contain the latent factors for each reviewer i, and $\vec{q}_j$, corresponding to each paper j. The factors are learned by minimizing the associated squared error function, using stochastic gradient descent.

$$\mathbf{P}_{ij} = \mu + \tau_i + \rho_j + \vec{p}_i \cdot \vec{q}_j \tag{3.4}$$

The number of latent factors f is a key parameter of this model. The dot product of $\vec{p}_m$ and $\vec{q}_n$ (for any specific reviewer m and paper n) provides a measure of similarity between m and n, based on the latent factor expansion. The resulting average test RMSE is slowly decreasing when increasing the dimensionality of the latent factor space. For example, for $f = 50$ it is **0.6240**, and for $f = 100$ it is **0.6234**. We use $f = 100$ throughout the remainder of the thesis, as it provides a good balance of performance and computation time for our data.

Number of factors	RMSE
f = 10	0.6258
f = 30	0.6249
f = 50	0.6240
f = 75	0.6237
f = 100	0.6234

3.1.3 Subject categories

In a typical conference submission process, authors are requested to denote primary and secondary categories appropriate for their papers. Likewise, reviewers are asked to indicate their interest along the same categories. It would be desirable to match reviewers with papers lying within their area of expertise. For ICDM'07, reviewers and papers are matched based on entered associations to 26 chosen topics or categories. The associated term makes use of three constant factors: a weight constant w_c, and relational constants θ_{ic} and σ_{jc} for all reviewers i, papers j, and categories c.

It is plausible that a mutual interest in category A will strongly link a reviewer to a paper, while a mutual interest in category B is less influential. For each category c, (w_c) indicates the significance of the category in linking a reviewer; again, these constants are learned automatically from the data.

Constants θ_{ic} and σ_{jc} relate reviewer i and paper j to category c as follows:

$$\theta_{ic} = \begin{cases} 1, & \text{if reviewer } i \text{ expressed interest in } c; \\ -0.5, & \text{if reviewer } i \text{ had lack of interest in } c; \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

$$\sigma_{jc} = \begin{cases} 1, & \text{if } c \text{ is a primary topic of paper } j; \\ 0.5, & \text{if } c \text{ is a secondary topic of paper } j; \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

These constants were provided with the ICDM'07 data; other conferences may provide different constants as desired. The complete term for this data source is:

$$\sum_{c=1}^{26} \theta_{ic} \cdot \sigma_{jc} \cdot w_c \tag{3.7}$$

This leads to a model which measures the interaction between reviewers and papers based on the association of the respective entered categories; adding this term to our model produces:

$$\mathbf{P}_{ij} = \mu + \tau_i + \rho_j + \vec{p_i} \cdot \vec{q_j} + \sum_{c=1}^{26} \theta_{ic} \cdot \sigma_{jc} \cdot w_c \tag{3.8}$$

The resulting average test RMSE of the model is: **0.6197**.

3.1.4 Paper & reviewer similarity metrics

In this model, we compute paper/paper similarity based on a vector space analysis of the paper abstracts. Cosine distance is an often used metric; a variant is the square of this distance, which has proven useful in various applications (e.g., see [38]). The squared cosine distance better contrasts the higher similarities against the lower ones. This models a reviewer's preferences in terms of their preferences for similar papers, as determined by our vector space model of paper abstracts. If s_{jk} is the square of the cosine distance between papers j and k, the model is:

$$\mathbf{P}_{ij} = \mu + \tau_i + \rho_j + \gamma \cdot \frac{\sum_{k \in S(i)} s_{jk} \mathbf{D}^{prefs}_{ik}}{\alpha + \sum_{k \in S(i)} s_{jk}} \tag{3.9}$$

where $s_{jj} = 0$, $S(i)$ contains all papers scored by reviewer i, and α is a small (regularization) constant to penalize cases where the average has very low support (i.e., where no similar paper was bid on by i). The parameter γ is the overall weight of the term, learned as part of the optimization algorithm. The resulting average test RMSE of this model is **0.6109**.

A very similar construction provides a reviewer/reviewer term, using the number of commonly co-authored papers t_{ij} as recorded in DBLP [25] (more sophisticated choices, perhaps based on published abstracts or papers of the reviewers, are a matter for future consideration):

$$\mathbf{P}_{ij} = \mu + \tau_i + \rho_j + \varsigma \cdot \frac{\sum_{l \in T(j)} t_{il} \mathbf{D}^{prefs}_{lj}}{\beta + \sum_{l \in T(j)} t_{il}} \tag{3.10}$$

Overall, the resulting RMSE of the reviewer/reviewer model combined with global effects is **0.6262**, thus offering less accuracy than the paper/paper model. In other settings, where higher quality reviewer/reviewer similarities are available, the relative merit of the model may increase.

Integrating paper/paper and reviewer/reviewer terms yields:

$$\mathbf{P}_{ij} = \mu + \tau_i + \rho_j + \vec{p}_i \cdot \vec{q}_j + \sum_{c=1}^{26} \theta_{ic} \cdot \sigma_{jc} \cdot w_c + \gamma \cdot \frac{\sum_{k \in S(i)} s_{jk} \mathbf{D}^{prefs}_{ik}}{\alpha + \sum_{k \in S(i)} s_{jk}} + \varsigma \cdot \frac{\sum_{l \in T(j)} t_{il} \mathbf{D}^{prefs}_{lj}}{\beta + \sum_{l \in T(j)} t_{il}} \tag{3.11}$$

All parameters are learned simultaneously by minimizing the associated squared error on the training set. This is our final prediction rule, which delivers an average test RMSE of **0.6015**. In the following section, we show how these predictions are transformed and merged with existing preferences.

3.2 Balancing predictions & preferences

The data source terms described above can be combined into a number of different models. This thesis uses the model in equation (3.11), which is a combination of global effects, categories, paper/paper, and latent factor terms from Section 3.1. We use our prediction model to find bids for all potential assignments. Before generating assignments using equation (2.3), it is important to balance the preference scale of various reviewers. Some reviewers tend to give mostly high bids, while others give medium to low bids. There are infinitely many ways to perform this balancing, depending on the specifics of the conference in question. Based on experimental results, we suggest the following alternative per-reviewer normalization strategies:

1. Subtract the per-reviewer mean from each prediction to find the **residual** preference for each potential assignment combination. (Henceforth dubbed as **Resid**.)

2. Calculate **normalized** preferences for each reviewer, so that the sum of each reviewer's predicted preferences is 1. (Henceforth dubbed as **Norm**.)

3. Bids for each reviewer are sorted and assigned an integral **rank** based on position in the sorted list, and normalized so the highest rank is 0.5. (Henceforth dubbed as **Ranked**.)

4. The sorted list from the 'Ranked' method above is divided into **quartiles**, and assigned a value 0, 1/4, 1/2, or 3/4 based on the bin into which each value falls. (Henceforth dubbed as **Quart**.)

After learning these models, we calculate predicted values for each possible assignment. To incorporate the effects of all of the relatively few preferences that are known, these predictions are

calculated 100 times using different randomly selected test-train splits, and the mean predictions for each potential reviewer/paper assignment are stored in matrix $\mathbf{P}$. These results are then combined with the original preferences $\mathbf{D}^{prefs}$ in an attempt to transform the partial ordering imposed on the preferences by $\mathbf{D}^{prefs}$ into a total ordering $\mathbf{D}$ for all reviewers i and papers j:

$$\mathbf{D}_{ij} = \mathbf{D}_{ij}^{prefs} + \mathbf{P}_{ij} \qquad \forall i, j. \tag{3.12}$$

In the next chapter, we evaluate these models against conference data from ICDM'07.

Chapter 4

Evaluation

Our modified total ordering of potential assignments **D** as defined in equation (3.12) is suitable for use in many of the assignment algorithms cited in Chapter 2. The constraint-based optimization introduced by Taylor [40] and cited in Section 2.2, for example, provides a linear program to generate assignments from a matrix of ratings similar to our **D** matrix. It also documents an evaluation criterion called 'affinity', which essentially considers the sum of the values from **D** corresponding to assignments made by the algorithm.

We have already demonstrated the ability of our modeling to capture reviewer/paper preferences. But do the improved models lead to better assignments? In other words, does Taylor's LP assignment algorithm leverage the improved modeling of preferences in ways that improve end-assignment quality? The key distinction is between *preferences* versus *assignments*, an aspect that has not been emphasized in prior recommender systems research.

The primary questions we seek to investigate are:

1. Do our preference models lead to higher quality assignments, based on the original evaluation criterion (affinity)?

2. Do our preference models lead to assignments with improved relevance using metrics based on additional data incorporated into our models? For example, do the assignments show a greater degree of topical relevance?

4.1 Data preparation

We study these issues in the context of the IEEE ICDM'07 conference data as described below. Data from real conferences is quite rare and difficult to obtain (acknowledged also in [30]), and in the future we hope that more datasets will become available to boost recommender systems research in conference management.

The dataset used in this thesis comes from the Seventh IEEE International Conference on Data Mining (ICDM'07) held in Omaha, NE, USA (used here with permission). The originally supplied

preference matrix ($\mathbf{D}^{prefs}$) is sparse: 529 papers, 203 reviewers, and only 6,267 bids (out of 107,387 potential assignments of a paper to a reviewer); 1587 actual assignments fills the requirement of three assignments per paper. This means that a reviewer bids on about 31 papers on average, while a paper receives less than 12 bids on average. We aim to assign each paper to 3 different reviewers, meaning about 9 different papers per reviewer; in some cases not enough bids exist to assign only papers with known bids to certain reviewers.

4.1.1 Preference value scale

Each value reflects a bid placed on a paper by a reviewer, with numerical values between 0 and 5, indicating preferences as defined by:

Preference value	Expressed preference
5	High (want to review)
4	OK to review
3	No preference expressed
2	Low
1	No (do not want to review)
0	Conflict of interest

Regardless of the chosen balancing scheme, we add the modified predicted value to the original preferences; unknown values in the original preference matrix are given a value between the 'OK' and 'Low' values. This forms our final input matrix $\mathbf{D}$, which we feed into Taylor's optimization algorithm. Taylor [40] does note that the chosen value scale can have a large effect on assignments; the numbers were chosen to impose a partial ordering on the potential assignments consistent with the expressed preference descriptions.

4.1.2 Mining conflict of interest data

One of the challenges of representing conflict of interest data for potential reviewers is the definition of conflict itself, which is variable based on the specific conference involved. As discussed earlier, there can be many sources of conflict to be considered. The list of possible sources includes affiliation with the same university or organization as an author, past co-authorship (with or without a specified expiration, after which no conflict is in effect), and personal familiarity with an author. It has even been suggested that two people with a common co-author ("friend of a friend") should be considered as having a conflict of interest. Obviously, most (if not all) academic conferences consider it a conflict of interest to review one's own paper.

Collecting this type of data presents its own set of challenges. For small conferences, it may be possible to manually screen the necessary people for potential conflicts, but this quickly becomes infeasible as the size of the conference (number of reviewers, papers, and authors) increases. Many conferences simplify the associated data gathering by asking reviewers to specify their own conflicts, based on agreed upon guidelines. While this streamlines the process, polling for conflicts

has its own set of drawbacks. For example, some reviewers may interpret the guidelines differently than others, leading to possible inconsistencies in the data. Consider reviewers A and B that each have papers submitted to the same conference, and reviewer A indicates a conflict with reviewing B's paper, but reviewer B reports no such conflict with reviewer A's paper. In the event of this kind of non-reflexive relationship of conflicts, the best course of action (or inaction) isn't immediately clear, without a specific guideline set forth by the conference.

In this thesis, conflict of interest (COI) data is represented as the binary matrix $\mathbf{D}^{coi}$; given reviewer i and author l, $\mathbf{D}^{coi}_{il}$ is 1 if a conflict exists, or 0 otherwise. Note that the definition is not intrinsically reflexive, so it is possible to model one-way conflicts if they exist. Also, as detailed above, it is quite possible to merge additional information (such as affiliation data) into $\mathbf{D}^{coi}$ in a way that allows a ranking among sources, with some taking precedence over others.

DBLP [25] is a web-based bibliographical database. It contains paper citations for many computer science journals and conference proceedings. This type of data is of great utility for discovering co-authors to be considered for the $\mathbf{D}^{coi}$ matrix. Rodriguez et al. [36] propagate a particle swarm across the co-authorship network of involved papers using DBLP, for the purpose of finding potential reviewers. A mentioned side benefit is the discovery of conflicts of interest based on direct co-authorship.

By extending this basic concept, one can discover potential conflicts involving more complex relationships in the co-author graph. For instance, a crawler program written for this research automatically downloads the DBLP citations page for a given author. The program proceeds by collecting all co-authors listed in the author's DBLP entry, and optionally discarding co-authors from works preceding a certain year. This automatic collection of co-authors greatly reduces the burden (usually on the conference chair or topic chairs) of producing conflict of interest data.

Criteria	Number of members
Found unmodified name in DBLP	162
Inconsistent initials, or Unicode characters in name	34
Spelling or miscellaneous errors in name	7
Total committee members	203

The automatic DBLP crawler was run for data from the International Conference for Data Mining (ICDM) in 2007 [34]. The crawler yielded immediate matches for 162 of 203 program committee members. Of the 41 non-matches, 34 were due to either inconsistent use of middle initial display, or the use of Unicode or international characters in the names. Both of these issues have potential to be corrected automatically by more advanced crawler logic, leaving 7 out of 203 that needed manual disambiguation by a human operator, due to spelling or miscellaneous errors. All of the committee members were found in DBLP after this disambiguation, which provided a complete set of conflict of interest data in a matter of hours.

Additionally, the crawler can consider potential transitive conflicts with an arbitrary number of indirections; more plainly, it can reconstruct multi-level connections between authors and potential reviewers. For example, a connection with one level of indirection would be a reviewer and an author having both co-authored (potentially different) papers with a third party. This simple example is extensible to multiple levels of indirection (or degrees of separation) in a co-author graph,

though each level greatly increases the number of accesses to DBLP. For direct co-authorship, only the program committee member pages need to be considered and crawled, but for indirect connections *all* co-authors for each committee member must be considered and crawled, since the third party connection between two committee members may not be on the committee.

The crawler could be further improved in a number of ways. First, additional logic could be added to automatically find DBLP entries for committee members with middle initials or names involving international characters. The latter members can be disambiguated using Unicode translation tables for such characters, while the former group might require additional work to gather any potential middle names or initials for all committee members. Also, the run time of the crawler is limited by access times to the DBLP website, and increased by the small gap between accesses granted as a courtesy to the website's maintainer. Faster run-time is possible by running against a recent copy of the DBLP data stored locally. Such a copy is periodically provided by DBLP, and would be sufficient for the purposes of mining conflict of interest data.

4.2 Assignment performance

We already have a preference value-based objective in the affinity metric defined by Taylor. To assess the topical relevance of the assignments, we evaluate them in terms of the mappings between papers/reviewers and subject categories. For every paper to reviewer assignment, we compute the dot product of the category vector of the paper with the category vector of the reviewer, and sum these dot products over the assignments made. Specifically paper/subject scores are recorded on a 2/1/0 scale (primary versus secondary versus neither) and reviewer/subject scores are recorded on a 1/-1/0 scale (interest versus conflict versus neither). In our dataset here, every paper has exactly one primary and one secondary category and hence the dot product can yield a number between -3 (reviewer has a conflict with both primary and secondary paper categories) and 3 (reviewer has interest in both paper categories). While other topical measures are certainly possible, the dot product method captures the relevance or 'on-topicness' of assignments made to each reviewer. Our four learned models are presented and compared to each other, as well as to the original Taylor LP-based assignments using only $\mathbf{D}^{prefs}$ as input:

	# of assigns	Affinity sum	Mean Affinity	Topical sum	Mean Topical
Taylor	1577	7192	4.5606	1869	1.1852
Resid, $f = 0$	1575	7168	4.5511	2140	1.3587
Resid, $f = 100$	1587	7222	4.5507	2117	1.3340
Norm, $f = 0$	1576	7189	4.5615	2112	1.3401
Norm, $f = 100$	1587	7222	4.5507	2101	1.3239
Ranked, $f = 0$	1579	7196	4.5573	2137	1.3534
Ranked, $f = 100$	1587	7220	4.5495	2097	1.3214
Quart, $f = 0$	1488	6926	4.6546	1977	1.3286
Quart, $f = 100$	1510	6991	4.6298	1945	1.2881

Fig. 4.1 depicts the results in terms of percentage improvement over the baseline Taylor approach (i.e., where only the original preferences without any additional data were input to the LP). Our

models are all evaluated using 100 latent factors, and also using 0 latent factors (essentially removing the latent factor term from consideration). Note that the latent factor term with 100 factors improves performance in all cases; therefore, we standardize our discussion for the remainder of the thesis using models with 100 latent factors. Note also that the topical evaluation metric shows a measurable improvement of assignments made using most of our learned prediction models. Since our new models take topical relevance into account, this is not unexpected. However, we accomplished this topical optimization without significantly degrading the Taylor algorithm's original 'affinity sum' objective; in fact, several models considered here slightly improve this objective as well.

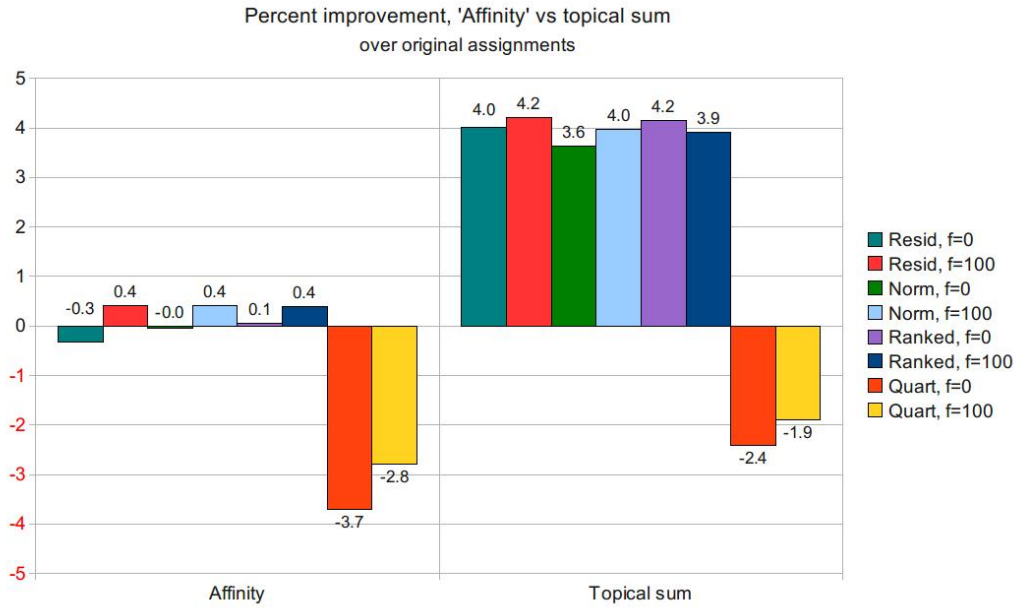

Figure 4.1: Performance comparison of various new models against unmodified assignments.

The Quart models do not perform as well as the others; it seems these methods do not provide enough additional ordering over the original preferences. Recall that the Quart models categorize the predictions in 4 'bins' of values; the greater dispersion of values provided by the other three prediction models provides additional flexibility in meeting the constraints, resulting in better performance.

Based on these numbers, the Resid and Norm models provide the most promising results, performing well in both affinity and topical evaluation objectives. Detailed improvement of these two models over the original preferences based solely on $\mathbf{D}^{prefs}$ is shown in Fig. 4.2.

4.3 Prediction quality

The common train-test split methodology is also useful for assessing prediction quality. In this section, both the prediction algorithm (3.11) and the assignment algorithm (2.3) cannot see the

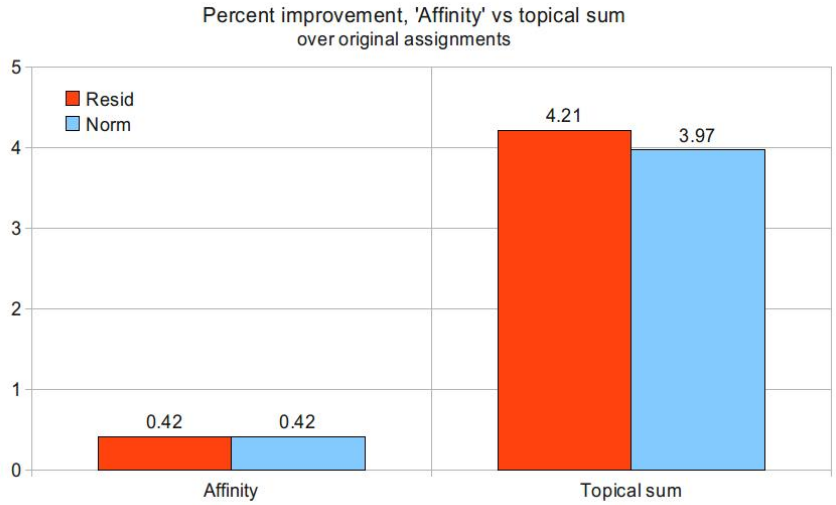

Figure 4.2: Topical relevance of assignments made with our approach versus Taylor's original formulation.

given preferences within the test set. Clearly, the elimination of the test set's preferences limits the flexibility of the assignment algorithm, as it has a lower number of favorable preferences from which to choose. However, the prediction model fills this gap by providing estimates to all missing preferences, including those in the test set. This simulates the real life scenario, where the given reviewer bids (corresponding to the training set) are limiting the possibilities of assignment algorithm, but by revealing more bids to the algorithms (including the test set) they gain the flexibility to provide better assignments.

As the proportion of the test set increases, we take away more available preferences, which simulates an increasingly harsh assignment environment. Accordingly, we increased the test set proportion to 30%, 40%, and 50% of "known" preferences (higher than the 10% used in Section 4.2) to simulate such an environment. We employed a series of 20 random train-test splits, generated assignments, and evaluated assignment quality at each iteration. The baseline is Taylor's original algorithm, where all missing bids, including those in the test set, are treated as "unknowns." We compare this baseline against two of our prediction models, Resid and Norm, which performed well in Section 4.2.

We evaluate quality of assignments by their ability to make good use of the hidden bids in the test set. The results are presented in Table 4.3 and Figs. 4.3, 4.4, and 4.5. As illustrated here, the predominant number (58-75%) of test assignments made using the original preference matrix fall in the non-preferred ("No") category. On the other hand, when imputing the missing bids using either Resid or Norm, the balance completely changes in favor of higher quality preferences. Resid makes over 58% of test assignments out of the highest quality bids ("High"), and only 11-14% of test assignments are bad ("No"). Norm also shows significant improvement over the original preferences, but is not quite as good as Resid, a difference that should be further investigated over additional datasets. Overall we find the results strongly support our goal to increase assignment quality by providing more flexibility with additional bids from which to choose.

% assignments from test set	Test set size	'No'	'Low'	'OK'	'High'
Taylor	30%	58.5%	0%	36.6%	4.9%
Taylor	40%	74.4%	2.4%	12.2%	11.0%
Taylor	50%	71.1%	2.3%	12.5%	14.1%
Norm	30%	17.3%	3.0%	28.7%	51.0%
Norm	40%	13.0%	3.1%	23.2%	60.8%
Norm	50%	14.6%	3.6%	29.2%	52.6%
Resid	30%	11.8%	3.9%	26.0%	58.3%
Resid	40%	13.7%	2.3%	21.1%	63.0%
Resid	50%	11.9%	2.8%	27.1%	58.3%

Table 4.1: Percentage of assignments made from the 'unknown' test set members, broken down by actual known preference values. Our methods show improvement over the unmodified Taylor LP, assigning a much higher percentage of 'preferred' papers.

4.4 Analysis of predicted values

We experimented with different ways of merging preference values and predicted values; the best results were generally obtained when considering all predictions (see equation (3.12)). However, an alternate method is to consider predictions only for paper/reviewer pairs with unknown preferences:

$$\mathbf{D}_{ij} = \begin{cases} \mathbf{D}^{prefs}_{ij} + \mathbf{P}_{ij}, & \text{if } \mathbf{D}^{prefs}_{ij} = 3; \\ \mathbf{D}^{prefs}_{ij}, & \text{otherwise.} \end{cases} \tag{4.1}$$

Under this alternate method, the total ordering provided by $\mathbf{P}$ only applies to potential assignments with unknown preference values, and those with known preferences are considered equivalent within each preference category. This approach can be considered a hybrid method, presenting less of a change from the original assignments, which consider only $\mathbf{D}^{prefs}$. We performed a comparison (shown in Fig. 4.6) of the above evaluation criteria on variants of our Resid models which follow this hybrid combination equation (4.1).

The hybrid models show a degradation in topical sum performance that is perhaps to be expected (since topical factors are only considered for unknown preference values in these models). However, they do show a slight improvement in affinity metric, which is interesting.

For all of our models, the *range* of predictions per reviewer is often quite large in comparison to the standard deviation. Fig. 4.7 shows an analysis of both range and standard deviation for $\mathbf{P}$, in this case for the Resid model with 100 latent factors. Apparently, most reviewers have a few outliers in their predicted values, thus widening the range. The reason for this pattern in predicted values is not entirely clear. It could indicate a tendency of reviewers to gravitate towards a single preference value; that is, most of the papers bid on by a reviewer are given the same preference value. As a consequence, the values in $\mathbf{P}$ are tightly clustered around the reviewer's

mean preference value. Situations where a reviewer deviated from that mean value would result in outliers, as the prediction model attempts to fit the known preferences.

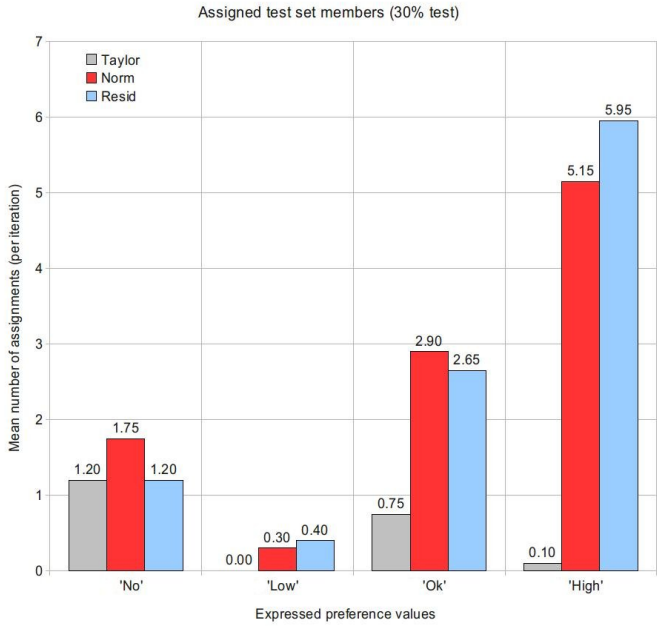

Figure 4.3: Evaluating the assignments made by the unmodified Taylor algorithm and the new preference models w.r.t. reviewers' four categories of preferences, using a 70-30 test-training set split, averaged across 20 iterations. Mean assignments per iteration are indicated above each bar.

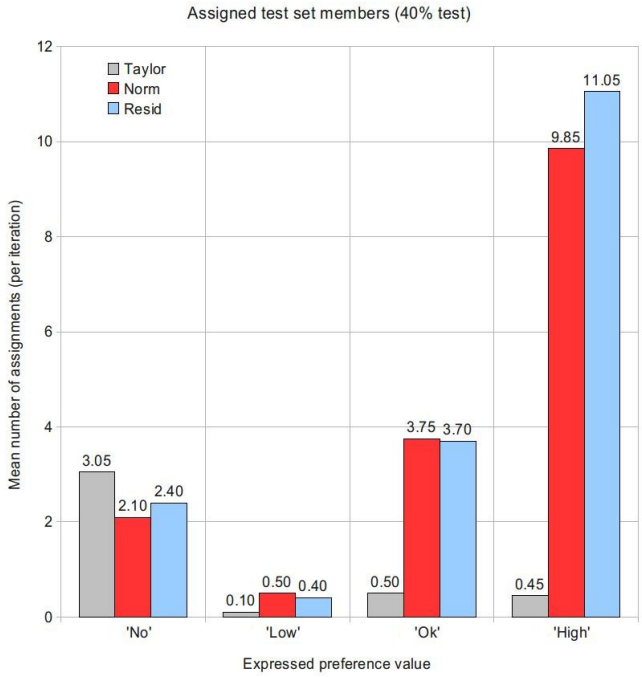

Figure 4.4: Evaluating the assignments made by the unmodified Taylor algorithm and the new preference models, using a 60-40 test-training set split, averaged across 20 iterations. Mean assignments per iteration are indicated above each bar.

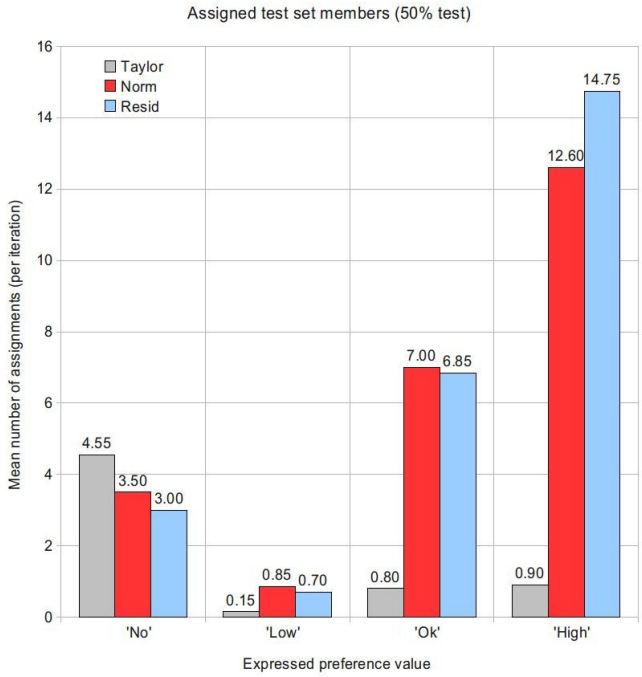

Figure 4.5: Evaluating the assignments made by the unmodified Taylor algorithm and the new preference models, using a 50-50 test-training set split, averaged across 20 iterations. Mean assignments per iteration are indicated above each bar.

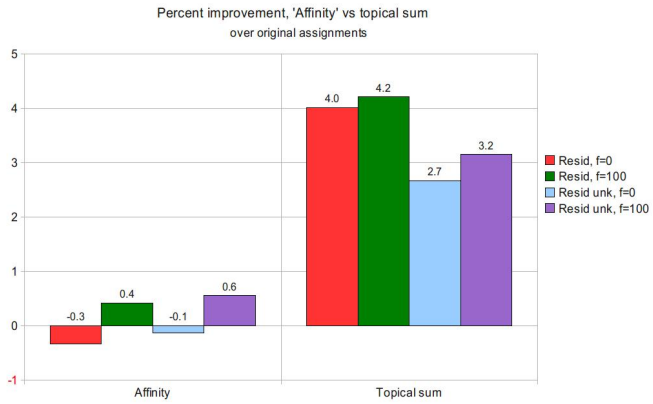

Figure 4.6: Comparison of one of our chosen models to a hybrid model which modifies only unknown preferences (labeled 'unk').

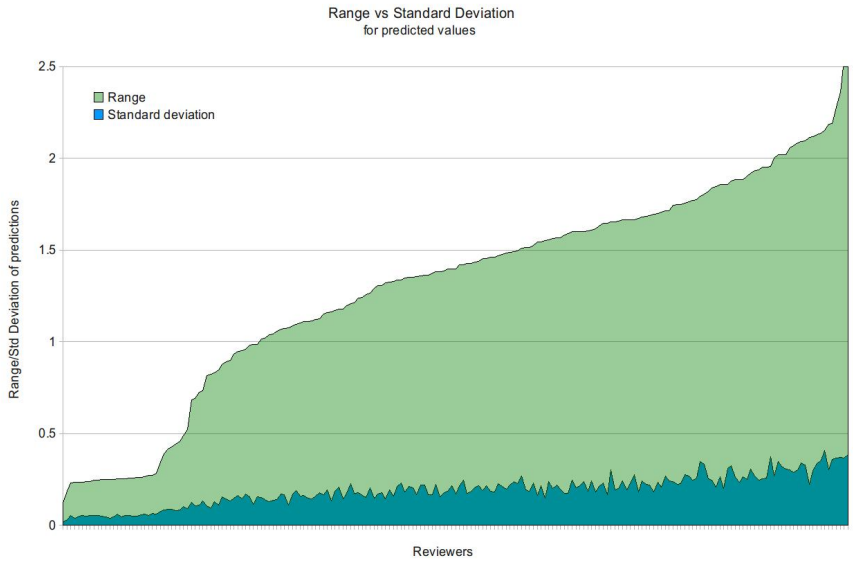

Figure 4.7: Prediction range and standard deviation per reviewer. While the range is often relatively large, predicted values typically appear to cluster close together, indicating outliers.

Chapter 5

Conclusion & Future Work

We have investigated the modeling of reviewer/paper preferences within a conference management system. The very limited data, typical to this context, requires identifying and exploiting multiple sources of information within our recommendation models. The proposed models provide improved predictions of reviewer preferences. More importantly, we showed how a better modeling of such preferences can lead to improvements in actual review assignments. Encouraging experimental results demonstrate that a better modeling can be well worth the effort in ensuring satisfaction of conference program committee reviewers. A key question for future work is to provide theoretical justification for the empirical evidence presented here. We also intend to field the recommendation capabilities presented here in a real conference management system and gain further insights into the issues involved.

The research associated with this thesis touches on a number of areas that could be developed further. The analysis of predicted values in Section 4.4 provides ample opportunity for further research into our predicted data models. Mining data from the web will continue to be more and more prevalent as the availability and quality of online information (and associated formats) improve. This provides a good source of supporting input data in cases when existing data is sparse or missing. The connection between frequent itemsets and bicliques shows promise as a measure of local connectivity.

One consequence of using our predictions to refine the existing partial ordering imposed by known preferences is that preference considerations still dominate in the assignments. Further study is needed on increasing the contributions from supporting data sources to the point where contributions from these sources can override the preference-derived partial ordering in certain cases.

Bibliography

[1] Microsoft Conference Management Toolkit (CMT).
`http://cmt.research.microsoft.com/cmt/`, June 2009.

[2] MyReview, a web-based conference management system.
`http://myreview.intellagence.eu/`, June 2009.

[3] START: Submission Tracking and Review Toolset.
`http://www.softconf.com/`, June 2009.

[4] The BYU conference management system.
`http://dagwood.cs.byu.edu/PaperReview`, June 2009.

[5] The ConfMan software.
`http://www.ifi.uio.no/confman/ABOUT-ConfMan/`, June 2009.

[6] The EasyChair software.
`http://www.easychair.org/`, June 2009.

[7] The OpenConf conference management system.
`http://zakongroup.com/technology/openconf.shtml`, June 2009.

[8] C. Basu, H. Hirsh, W. Cohen, and C. Nevill-Manning. Technical paper recommendation: a study in combining multiple information sources. *Journal of Artificial Intelligence Research*, 14:231–252, 2001.

[9] R. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proc. of the 7th IEEE International Conference on Data Mining*, pages 43–52, August 2007.

[10] S. Benferhat. Conference paper assignment. *International Journal of Intelligent Systems*, 16(10):1183–1192, 2001.

[11] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[12] N. Di Mauro, T. M. Basile, and S. Ferilli. GRAPE: an expert review assignment component for scientific conference management systems. In *Proc. of the 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 789–798, 2005.

[13] S. T. Dumais and J. Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *Proc. of the 15th ACM International Conference on Research and Development in Information Retrieval*, pages 233–244, June 1992.

[14] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35:61–70, 1992.

[15] J. Goldsmith and R. H. Sloan. The AI conference paper assignment problem. In *Pref. Handling for Artficial Intelligence, Papers from the AAAI Workshop*, July 2007.

[16] P. Halvorsen, K. Lund, V. Goebel, T. Plagemann, T. Preuss, and H. Koenig. Architecture, implementation, and evaluation of ConfMan: integrated WWW and DBS support for conference organization. In *Technical Report I-1998.016-R*. University of Oslo, Norway, December 1998.

[17] D. Hartvigsen, J. C. Wei, and R. Czuchlewski. The conference paper-reviewer assignment problem. *Decision Sciences*, 30(3):865–876, 1999.

[18] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, 2004.

[19] J. Hopcroft and R. Karp. An $n^{2.5}$ algorithm for maximum matching in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, December 1973.

[20] M. D. Humphries and K. Gurney. Network "small-world-ness": a quantitative method for determining canonical network equivalence. *PLoS ONE*, 3(4):e0002051, April 2008.

[21] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[22] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. of the 14th ACM International Conference on Knowledge Discovery and Data Mining*, pages 426–434, August 2008.

[23] S. Krishnamurthi, P. W. Hopkins, J. Mccarthy, P. T. Graunke, G. Pettyjohn, and M. Felleisen. Implementation and use of the PLT scheme web server. *Higher Order and Symbolic Computation*, 20(4):431–460, 2007.

[24] H. W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[25] M. Ley and P. Reuther. Maintaining an online bibliographical database: the problem of data quality. In *Extraction et Gestion des Connaissances*, pages 5–10, January 2006.

[26] J. Li, H. Li, D. Soh, and L. Wong. A correspondence between maximal complete bipartite subgraphs and closed patterns. In *Proc. of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 146–156, October 2005.

[27] J. Li, G. Liu, H. Li, and L. Wong. Maximal biclique subgraphs and closed pattern pairs of the adjacency matrix: a one-to-one correspondence and mining algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1625–1637, 2007.

[28] S. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI Extended Abstracts*, pages 1097–1101, April 2006.

[29] S. Milgram. The small world problem. *Psychology Today*, 1(61):60–67, 1967.

[30] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *Proc. of the 13th ACM International Conference on Knowledge Discovery and Data Mining*, pages 500–509, August 2007.

[31] M. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[32] O. Nierstrasz. Identify the champion. In *Pattern Languages of Program Design*, volume 4, pages 539–556, December 1999.

[33] R. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proc. of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 437–444, August 2001.

[34] N. Ramakrishnan, O. Zaiaine, Y. Shi, C. Clifton, and X. Wu, editors. *Proc. of the 7th IEEE International Conference on Data Mining*, October 2007.

[35] P. Rigaux. An iterative rating method: application to web-based conference management. In *Proc. of the ACM Symposium on Applied Computing*, pages 1682–1687, March 2004.

[36] M. A. Rodriguez, J. Bollen, and H. Van de Sompel. The convergence of digital libraries and the peer-review process. *Journal of Information Science*, 32(2):149–159, 2006.

[37] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proc. of the 24th ACM International Conference on Machine Learning*, pages 791–798, June 2007.

[38] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. of the 10th International Conference on the World Wide Web*, pages 285–295, May 2001.

[39] R. Snodgrass. Summary of conference management software.
`http://www.acm.org/sigs/sgb/summary.html`, June 2009.

[40] C. J. Taylor. On the optimal assignment of conference papers to reviewers. Technical Report MS-CIS-08-30, University of Pennsylvania, 2008.

[41] R. van de Stadt. CyberChair: A web-based groupware application to facilitate the paper reviewing process. `http://cyberchair.org/`, June 2009.

[42] D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, June 1998.

[43] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *Proc. of the 29th ACM International Conference on Research and Development in Information Retrieval*, pages 178–185, August 2006.

[44] W. Xi, E. A. Fox, W. Fan, B. Zhang, Z. Chen, J. Yan, and D. Zhuang. Simfusion: measuring similarity using unified relationship matrix. In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval*, pages 130–137, 2005.

[45] D. Yarowsky and R. Florian. Taking the load off the conference chairs: towards a digital paper-routing assistant. In *Proc. of the Joint SIGDAT Conference on Empirical Methods in NLP and Very-Large Corpora*, June 1999.

[46] M. J. Zaki and C.-J. Hsiao. Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):462–478, 2005.