Recommender Systems for the Conference Paper Assignment Problem

Don Conry
Virginia Tech
Blacksburg, VA, USA
dconry@cs.vt.edu

Yehuda Koren
Yahoo! Research
Haifa, Israel
yehuda@yahoo-inc.com

Naren Ramakrishnan
Virginia Tech
Blacksburg, VA, USA
naren@cs.vt.edu

ABSTRACT

We present a recommender systems approach to conference paper assignment, i.e., the task of assigning paper submissions to reviewers. We address both the modeling of reviewer-paper preferences (which can be cast as a learning problem) and the optimization of reviewing assignments to satisfy global conference criteria (which can be viewed as constraint satisfaction). Due to the paucity of preference data per reviewer or per paper (relative to other recommender systems applications) we show how we can integrate multiple sources of information to learn reviewer-paper preference models. Our models are evaluated not just in terms of prediction accuracy but in terms of end-assignment quality. Using a linear programming-based assignment optimization, we show how our approach better explores the space of unsupplied assignments to maximize the overall affinities of papers assigned to reviewers. We demonstrate our results on real reviewer bidding data from the IEEE ICDM 2007 conference.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*; J.4 [**Computer Applications**]: Social and Behavioral Sciences

General Terms

Algorithms, Human Factors

Keywords

Recommender systems, collaborative filtering, conference paper assignment, linear programming

1. INTRODUCTION

Modern conferences are beset with excessively high numbers of 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.

The primary input to the conference paper assignment problem (CPAP) is a papers × reviewers matrix of 'bids', expressing interest or disinterest of reviewers to review specific papers. The goal is to construct a set of reviewing assignments taking into account reviewer capacity constraints, adequate numbers of reviews for papers, expertise modeling, conflicts of interest, and other global conference criteria.

There are three key differences between traditional recommender applications and the CPAP problem. (i) In a traditional recommender, recommendations that meet the needs of one user do not affect the satisfaction of other users. In CPAP, on the other hand, multiple users (reviewers) are bidding to review the same papers and hence there is the possibility of one user's recommendations (assignments) affecting the satisfaction levels (negatively) of other users. Hence the design of reviewer preference models must be posed and studied in an overall optimization framework. (ii) 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 aiming to satisfy multiple users. Finally, (iii) recommender systems are used to working with sparse data but the amount of 'signal' available to model preferences in the CPAP domain is exceedingly small; hence we must integrate multiple sources of information to build strong preference models.

We organize our framework into two stages: 'growing' the given bids by adapting recommendation techniques to predict unknown reviewer-paper preferences, and identifying a good assignment by optimizing conference criteria. Other approaches to CPAP (e.g., [1]) are surveyed elsewhere [2]. We apply our framework on bids and auxiliary information (see Fig. 1) gathered from the *7th IEEE Intl. Conf on Data Mining (ICDM'07)* for which the third author was a program chair. Similar scope datasets from other conferences are not publicly available (also acknowledged in [5]) and we hope our research will spur greater availability. (The Cyberchair system used by the ICDM series has expressed interest in implementing our approach and we plan to approach Easychair and other CMSs as well.) We emphasize that all datasets were anonymized before the modeling and analysis steps conducted here.

2. MODELING REVIEW PREFERENCES

We are given *ratings* (henceforth, interchangeable with *preferences*) between m reviewers and n papers. (Recall that

RecSys'09, October 23–25, 2009, New York, New York, USA.
Copyright 2009 ACM 978-1-60558-435-5/09/10 ...$10.00.

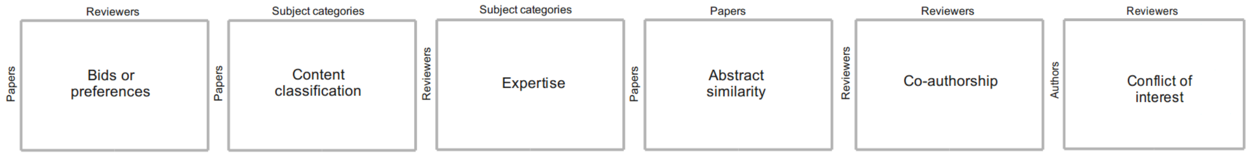

Figure 1: Data used in this paper for building paper-reviewer preference models.

these ratings are really bids/signs of interest to review papers, not the actual ratings reviewers assign to papers after reading and evaluating them.) A rating r_{ui} indicates the preference by reviewer u of paper i, where high values mean stronger preferences. Usually the vast majority of ratings are unknown, e.g., the ICDM data involves 529 papers, 203 reviewers, and only 6267 bids. In ICDM'07, the given bids are between 1 and 4, indicating preferences as follows: 4= "High", 3="OK", 2="Low" and 1="No" and we aim to make predictions in the same space.

We distinguish predicted ratings from known ones, by using the notation $\hat{r}_{ui}$ for the predicted value of r_{ui}. To evaluate the models we assess RMSE over 100 random 90-10 training-test splits. We hasten to add that we do not advocate the myopic view of RMSE [4] as the primary criterion for recommender systems evaluation. We use it in this section primarily due to its convenience for constructing direct optimizers. In the next section we will evaluate performance according to criteria more natural to CPAP. 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 model we learn is of the form:

$$\begin{aligned}\hat{r}_{ui} &= \mu + b_u + b_i + p_u^T q_i + \sum_c \sigma_{ic}\theta_{uc}w_c \\ &+\gamma \frac{\sum_{j\in \mathrm{R}(u)} s_{ij} r_{uj}}{\alpha + \sum_{j\in \mathrm{R}(u)} s_{ij}} + \phi \frac{\sum_{v\in \mathrm{R}(i)} s_{uv} r_{vi}}{\beta + \sum_{v\in \mathrm{R}(i)} s_{uv}} \qquad (1)\end{aligned}$$

and we proceed to explain each of the terms below.

2.1 Baseline model

Much of the variability in the data is explained by global effects, which can be reviewer- or paper-specific. It is important to capture this variability by a separate component, thus letting the more involved models deal only with genuine reviewer-paper interactions. We model these global effects through the first three terms of Eq. 1, i.e., $\mu + b_u + b_i$. The constant μ indicates a global bias in the data, which is taken to be the overall mean rating. The parameter b_u captures reviewer-specific bias, accounting for the fact that different reviewers use different rating scales. Finally, the paper bias, b_i, accounts for the fact that certain papers tend to attract higher (or, lower) bids than others. We learn optimal values for b_u $(u = 1, \ldots, m)$ and b_i $(i = 1, \ldots, n)$, by minimizing the associated squared error function with just these three terms (along with some regularization to avoid overfitting). The resulting average test RMSE is **0.6286**.

A separate analysis of each of the two biases shows reviewer effect ($\mu + b_u$, with RMSE **0.6336**) to be much more significant than paper bias ($\mu + b_i$, RMSE **1.2943**) in reducing the error. This indicates a tendency of reviewers to concentrate all ratings near their mean ratings, which is supported by examination of the data.

While the baseline model could explain much of the data variability, as evident by its relatively low associated RMSE, it is useless for making actual assignments. After all, it gives all reviewers exactly the same order of paper preferences. Thus, we are really after the remaining unexplained variability, where reviewer-specific preferences are getting expressed. Uncovering these preferences is the subject of the next subsections.

2.2 A factor model

Latent factor models (e.g., [3]) comprise a common approach to collaborative filtering with the goal to uncover latent features that explain observed ratings. The premise of such models is that both reviewers and papers can be characterized as vectors in a common f-D space. The interaction between reviewers and papers is modeled by inner products in that space, the fourth term of Eq. 1. Here, $p_u \in \mathbb{R}^f$ and $q_i \in \mathbb{R}^f$ are the factor vectors of reviewer u and paper i, respectively. The resulting average test RMSE is slowly decreasing when increasing the dimensionality of the latent factor space. E.g., for $f = 50$ it is **0.6240**, and for $f = 100$ it is **0.6234**. Henceforth, we use $f = 100$.

2.3 Subject categories

While latent factor models automatically infer suitable categories, much can be learned by known categories attributed to both papers and reviewers. ICDM'07 submissions specify a number of predefined categories as primary and secondary topics for a given paper. We model the entered matching between paper i and category c by:

$$\sigma_{ic} = \begin{cases} 1 & c \in \text{primary}(i) \\ \frac{1}{2} & c \in \text{secondary}(i) \\ 0 & \text{otherwise} \end{cases}$$

The value assignment (1 for "primary", 0.5 for "secondary") is derived by cross validation and is quite intuitive. Similarly, we use the following for matching reviewers with their desired categories:

$$\theta_{uc} = \begin{cases} 1 & c \in \text{interest}(u) \\ -\frac{1}{2} & c \in \text{no\_interest}(u) \\ 0 & \text{otherwise} \end{cases}$$

Notice that in ICDM'07, reviewers could specify lack of interest (or inability to) review papers from certain categories (this is different from conflicts of interest, discussed later).

In the fifth term of Eq. 1, the weights w_c indicate the significance of each category in linking a reviewer to a paper, and are learnt automatically by minimizing the squared error on the training set. It is plausible that, e.g., a mutual interest in some category A, will strongly link a reviewer to a paper, while a mutual interest in another category B is less influential on papers choice. Table 1 depicts results of this analysis, showing differences in orders of magnitude in the ability of different categories to correctly predict associations of reviewers to papers. Note in particular that there is no obvious monotonic relationship between the weight imputed to categories and the number of papers/reviewers associated

Table 1: Subject categories, inferred weights, number of reviewers (with expertise in that category), and number of papers (assigned to the category). For brevity, only a few categories are shown.

Category	Weight (w_c)	# reviewers	# papers primary (secondary)
Healthcare, epidemic modeling, and clinical research	0.395121	31	7 (7)
Security, privacy, and data integrity	0.334821	23	12 (6)
Handling imbalanced data	0.284398	24	6 (10)
Mining textual and unstructured data	0.245319	66	38 (30)
Mining in networked settings: web, social and computer networks, and online communities	0.206318	62	44 (29)
Novel data mining algorithms in traditional areas (such as classification, regression, clustering, probabilistic modeling, and association analysis)	0.089248	91	147 (71)
Dealing with cost sensitive data and loss models	0.03453	12	4 (4)
Algorithms for new, structured, data types, such as arising in chemistry, biology, environment, and other scientific domains	0.006015	60	21 (25)

with the category. When adding subject categories to the baseline and factor models, the resulting RMSE is **0.6197**.

2.4 Paper-paper similarities

We inject paper-paper similarities into our models in a way reminiscent of item-item recommenders [6]. The building blocks here are similarity values s_{ij}, which measure the similarity of paper i and paper j. The similarities could be derived from the ratings data, but those are already covered by the latent factor model. Rather, we derive the similarity of two papers by computing the cosine of their abstracts. Usually we work with the square of the cosine, which better contrasts the higher similarities against the lower ones.

In the sixth term of Eq. 1, the set $\mathrm{R}(u)$ contains all papers on which u bid. The constant α is for regularization: it is penalizing cases where the weighted average has very low support, i.e. $\sum_{j\in\mathrm{R}(u)} s_{ij}$ is very small. In our dataset it was determined by cross validation to be 0.001. The parameter γ sets the overall weight of the paper-paper component. It is learnt as part of the optimization process (cross-validation could have been used as well). Its final value is close to 0.7. When this term is combined with the overall scheme, the RMSE drops down further to **0.6038**.

2.5 Reviewer-reviewer similarities

We craft reviewer-reviewer similarities s_{uv} analogously to paper-paper similarities, measured as the number of commonly co-authored papers as reported in DBLP. We point out that DBLP data might be incomplete, and co-authorship does not imply similarity of research interests. Nevertheless, our main contribution here is to show how to incorporate reviewer-reviewer similarities in Eq. 1 and more sophisticated ways to define s_{uv} can be readily plugged in. By integrating this factor, the RMSE is **0.6015**.

2.6 Conflicts of interest (CoI)

A final source of data is conflicts of interest for certain (paper, reviewer) combinations, e.g., the reviewer might be the former advisor of the author. Many conferences define what it means to have a CoI and solicit this information explicitly during the bidding phase. We do not aim to model/predict new CoIs but show in the next section how they are incorporated to avoid making erroneous assignments.

3. OPTIMIZING PAPER ASSIGNMENT

Our predicted preference matrix can now be supplied as input to any of the assignment algorithms discussed in [2]. We chose the Taylor algorithm [7] as a representative example because it was used during ICDM'07 and thus enables a baseline comparison with an approach that does not perform any preference modeling. It can incorporate global conference constraints such as the desired number of reviewers for each paper (k_p), and a desired maximum number of papers for each reviewer (k_r). (For ICDM'07, these values are 3 and 9, respectively.) Denoting the predicted ratings matrix as $\mathbf{R}$, the goal is to optimize the assignments matrix $\mathbf{A}$ [7]:

$$\begin{aligned}\operatorname*{argmax}_{\mathbf{A}} \quad & \text{trace}\left(\mathbf{R}^T\mathbf{A}\right) = \operatorname*{argmax}_{\mathbf{A}} \sum_u \sum_j \mathbf{R}_{uj}\mathbf{A}_{uj}, \qquad (2)\\ & \text{where} \qquad \mathbf{A}_{uj} \in [0,1] \qquad \forall u, j,\\ & \text{and} \quad \sum_j \mathbf{A}_{uj} \le k_p, \qquad \forall u,\\ & \text{and} \quad \sum_u \mathbf{A}_{uj} \le k_r, \qquad \forall j.\end{aligned}$$

Here, the objective criterion—trace $\left(\mathbf{R}^T\mathbf{A}\right)$—captures the global affinity of all reviewers across all their assigned papers. CoIs can be modeled by hardwiring the desired entries of $\mathbf{A}$ (to zero) and taking them 'out of play' in Eq. 2.

This integer programming problem 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 $\mathbf{A}$, and every column represents an edge [7]. This reformulation is a bit more complicated, but essentially renders the problem solvable via methods such as Simplex or interior point programming. In particular, as Taylor shows in [7], because the reformulated constraint matrix is *totally unimodular*, there exists at least one globally optimal assignment with integral (and due to the constraints, Boolean) coefficients.

4. EXPERIMENTAL RESULTS

We have already shown the ability of our modeling to better capture reviewer-paper preferences. But do the improved models translate into better assignments? Note the key distinction between *recommendations* and *assignments.* To evaluate assignment quality, we extend the train-test methodology from above. In other words, both the prediction algorithm and the assignment algorithm cannot see the originally given preferences within the test set. We use the training set to learn model (1), predict *all* ratings using this model, and feed these predictions as input to (2). While the resulting assignment will be spread across the training and test sets, we will specifically evaluate those made from the test set and determine whether the reviewer had rated them as 'No,' 'Low,' 'OK,' or 'High.' This methodology mimics the real life scenario where the given reviewer ratings (corresponding to the training set) are limiting the possibilities of the assignment algorithm, but by revealing more ratings through our prediction phase, we aim to 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 harsher assignment environment.

However, before using Taylor's model (2), it is important to balance the rating scale of various reviewers. For example, some reviewers are very enthusiastic and tend to give mostly high ratings, while others are more cautious and give low ratings. While our preference modeling captures such variance, it is unnecessary for the assignment phase since Taylor's model would concentrate only on reviewers with high ratings, which is undesirable. Thus, we suggest two alternative per-reviewer normalization strategies:

1. Subtract the per-reviewer mean from each predicted rating to find the **residual** rating for each potential assignment combination. (Henceforth dubbed as **Resid**.)
2. Calculate **normalized** ratings for each reviewer, so that the sum of each reviewer's predicted ratings is 1. (Henceforth dubbed as **Norm**.)

Regardless of the chosen normalization scheme, we add the normalized predicted rating to the original preferences (if it is part of the training data) or to the mean rating value (2.5) (for test data; recall that this is between the 'Ok' and 'Low' ratings). This forms our final input matrix **R**, which we feed into Taylor's optimization algorithm.

We evaluate many train-test splits, averaging 100 random trials for each split. The baseline is Taylor's original algorithm, where all missing ratings, including those in the test set, are treated as "unknowns." We compare this baseline against the two aforementioned alternatives, Resid and Norm, with an identical handling for missing ratings. Specifically, we look at the proportion of assignments from the test set that fall in the 'No,' 'Low,' 'OK,' and 'High' categories.

Table 2: Evaluating assignments: observe the dramatic improvement from the baseline (Taylor) to our methods (Norm and Resid).

Method	Test	Ratings			
	set	'No'	'Low'	'OK'	'High'
Taylor	30%	59.9%	0.1%	30.2%	8.5%
Taylor	40%	63.3%	1.4%	22.5%	12.7%
Taylor	50%	63.6%	2.6%	17.4%	16.4%
Norm	30%	16.5%	0.4%	25.3%	54.2%
Norm	40%	11.9%	5.1%	23.8%	59.2%
Norm	50%	13.2%	5.5%	25.2%	56.1%
Resid	30%	11.1%	3.2%	24.6%	61.0%
Resid	40%	10.9%	3.2%	24.5%	61.3%
Resid	50%	11.2%	4.0%	24.1%	60.6%

The results presented in Table 2 were fairly consistent across different test set proportions. As illustrated here, the predominant number (around 60-65%) of test assignments made using the original preference matrix (Taylor) fall in the unpreferred ("No") category, mirroring experiences during ICDM'07 organization[1]. On the other hand, when imputing the missing ratings using either Resid or Norm, the balance completely changes in favor of higher quality preferences. Resid makes about 60% of test assignments out of the highest quality ratings ("High"), and only about 12% of test assignments are bad ("No"). Norm is close, but not quite as good as Resid, a difference that should be further investigated over additional datasets. Overall we find that the results strongly support our contention that assignment quality can be increased by providing more flexibility with additional ratings from which to choose.

5. DISCUSSION

Why does our approach work? Especially with harsh train-test splits? If we view a reviewer's preferences as a partial order over papers, we can think of our approach as 'straightening' out the partial order into a total order that is consistent with multiple sources of data. We intend to provide theoretical justification for our empirical results using this viewpoint. The second new aspect to our work is the integration of recommendation and optimization/constraint satisfaction. In the future we seek to study how recommenders help aid optimization routines by providing additional 'cues' or flexibilities in constraint satisfaction/search. Besides CPAP, this has applications to combined recommendation-optimization scenarios such as targeted marketing and advertising under resource constraints. Finally, to gain qualitative user feedback, we intend to field the recommendation/assignment capabilities presented here in a real conference management system and gain further insights into the issues involved.

6. ACKNOWLEDGEMENTS

We thank Prof. Xindong Wu, Steering Committee chair of IEEE ICDM for permission to use bidding and auxiliary data from ICDM'07.

7. REFERENCES

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

[2] D. Conry, Y. Koren, and N. Ramakrishnan. Recommender systems for the conference paper assignment problem. Technical report, arXiv: 0906.4044v1, 2009.

[3] T. Hofmann. Latent semantic models for collaborative filtering. *ACM TOIS*, 22:89–115, 2004.

[4] S. McNee, J. Riedl, and J. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI Extended Abstracts*, pages 1097–11101, 2006.

[5] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *Proc. KDD'07*, pages 500–509, 2007.

[6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW'01*, pages 285–295, 2001.

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

[1] The assignments were manually re-wired afterward.