Collaborative Future Event Recommendation

Einat Minkov*
Univ. of Haifa
Haifa, Israel
einatm@mis.haifa.ac.il

Ben Charrow†
Univ. of Pennsylvania
Philadelphia PA, USA
charrow@cs.upenn.edu

Jonathan Ledlie
Nokia Research
Cambridge MA, USA
jonathan.ledlie@nokia.com

Seth Teller
MIT CSAIL
Cambridge MA, USA
teller@csail.mit.edu

Tommi Jaakkola
MIT CSAIL
Cmabridge MA, USA
tommi@csail.mit.edu

ABSTRACT

We demonstrate a method for collaborative ranking of future events. Previous work on recommender systems typically relies on feedback on a particular item, such as a movie, and generalizes this to other items or other people. In contrast, we examine a setting where no feedback exists on the particular item. Because direct feedback does not exist for events that have not taken place, we recommend them based on individuals' preferences for past events, combined collaboratively with other peoples' likes and dislikes. We examine the topic of unseen item recommendation through a user study of academic (scientific) talk recommendation, where we aim to correctly estimate a ranking function for each user, predicting which talks would be of most interest to them. Then by decomposing user parameters into shared and individual dimensions, we induce a similarity metric between users based on the degree to which they share these dimensions. We show that the collaborative ranking predictions of future events are more effective than pure content-based recommendation. Finally, to further reduce the need for explicit user feedback, we suggest an active learning approach for eliciting feedback and a method for incorporating available implicit user cues.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

General Terms

Algorithms, Design, User Study

Keywords

Recommendation Systems, Collaborative Filtering

1. INTRODUCTION

Recommender systems aim to present items that are likely to be of interest to users. Such systems are widely implemented in industrial settings, including in e-commerce systems where the goal is to recommend products to users based on their purchase, viewing, or rating history. For example, the Netflix challenge [3] was introduced to test recommendation methods based on historical movie ratings provided by customers.

The general recommendation setting consists of a pool of items and a pool of users, where user feedback such as ratings are available for varying subsets of items. The goal is to use the partial rating history to predict ratings for the remaining items that the users have yet to experience. Methods used to solve this problem can be roughly categorized into *content-based* and *collaborative filtering* approaches. Content-based approaches leverage feature descriptions of items and seek to relate user preferences to those features. Collaborative filtering methods, in contrast, rely solely on the historical record of user ratings and similarities across users in order to fill-in ratings for the remaining items. The problem is often cast as a matrix completion problem and solved using matrix factorization methods [16]. The two approaches are largely complementary.

In this paper, we consider a recommendation system in which the information items are *events*. Event recommendation has various potential applications. Consider a location-tracking system, which records the user's whereabouts within a geographical region, or inside a building [15]. Given location coordinates, it is desirable to provide users with personalized location-based services. In this framework, event recommendation can bring to the user's attention relevant events that take place nearby. Another motivation is assisting event organizers; if historical preferences about potential attendees (the users) are available, recommender systems can be used to predict the overall interest in the event. Such predictions can be used to guide resource management as well as to identify potential conflicts between simultaneous events.

*A major part of this work has been conducted while affiliated with Nokia Research, Cambridge MA.

†A major part of this work has been conducted while affiliated with MIT CSAIL, and during an internship at Nokia Research, Cambridge MA.

CIKM'10, October 26–29, 2010, Toronto, Ontario, Canada.

Our contributions are:

- We present a content-based approach to recommending events to users based on users' past selections and feature descriptions of the events.
- We propose a collaborative extension of the basic approach, LowRank , by decomposing user parameters into shared and individual components.
- We evaluate our approaches through a user study involving about 90 users and 15 weeks worth of talk announcements.
- We provide a framework for actively eliciting relevant information from users so as to minimize the amount of explicit feedback needed to make accurate predictions.

The paper proceeds as follows. Section 1.1 defines event recommendation as a ranking problem, and describes its modeling using a pure content-based approach. In Section 2 we outline our collaborative method for event recommendation. Sections 3 and 4 describe the user study and event descriptions, respectively. We evaluate LowRank as compared to RankSVM in Section 5. Section 6 outlines an active learning method for eliciting relevant feedback from users that could further speed up LowRank . We review related work in Section 7. In Section 8, we conclude and discuss future work.

1.1 Event Recommendation

Informally, an event is an information item that is only valid for a short period of time. By the time one can expect user feedback on a specific event, that event is no longer relevant: an event recommendation system therefore has to recommend items for which no explicit feedback exists. This distinguishes events from other information items, like movies, for which some user feedback is directly available and continues to be useful. In general, one can approach event recommendation using existing content-based methods, relating event descriptions to user preferences. However, the quality of content-based recommendation is often highly dependent on the amount of user feedback available. We expect feedback about events to be relatively scarce; events are often topically diverse and new events may have low similarity to prior events. Modeling user preferences collaboratively may alleviate this data sparsity problem, by pooling together feedback from users with similar preferences.

In this paper, we present a collaborative approach to event recommendation. Each user has a parameter vector that relates their preferences to event descriptions. This is necessary in order to be able to recommend yet unseen events. However, in our case, we first map event descriptions to a low-dimensional "perceptual space". Intuitively, the low-dimensional representation of events captures how events vary perceptually to users and are shared across users. The user parameters are then associated with these coordinates rather than the original feature descriptions of events. The number of parameters required to capture individual user preferences is therefore considerably smaller than in a regular content-based approach.

We expect our approach, LowRank , to work best in a scenario where the available feedback about past events varies across users. For example, different subsets of users may have rated different past events. Such experience would help uncover the relations between the events, their low dimensional feature representations, and the associated user preferences.

In order to evaluate our approach, we have carried out an empirical user study of event recommendation. Specifically, the user study, conducted at the Massachusetts Institute of Technology (MIT) and at Carnegie Mellon University (CMU), concerns the recommendation of scientific talks[1] to users within their universities. The MIT group includes thirty users, who are mostly graduate students, research affiliates and faculty members at the Computer Science and Artificial Intelligence Laboratory (CSAIL). The CMU group includes graduate students of the computer science school. The study was designed to simulate a realistic setting, where user feedback becomes available after events take place and the task is to recommend upcoming events. We consider a ranking setting, where, in the beginning of every week, the talks scheduled for that week are ranked according to each user's preferences. The data used in the study involves real seminar announcements, following their original temporal distribution.

In our experiments, we evaluate recommendation performance for several sets of future talks, given varying amounts of feedback on past talks. We evaluate content-based recommendation methods using RankSVM [11]. Several sets of features are considered, including a traditional word-based model, as well as representing each talk in terms of its topic usage (see Section 4). We subsequently compare the performance of content-based ranking with our proposed LowRank method that supports collaborative ranking. The collaborative approach is shown to give superior performance.

Overall, our results show that it is possible to achieve good results on the task of scientific talk recommendation, which may be sufficient for practical applications. These results may be improved further by means of feature engineering.

Finally, we are interested in further reducing the amount of feedback required from each user. To this end, we suggest an active learning approach to this problem. While we do not give full results for active learning, our preliminary results are encouraging.

In summary, the main contribution of this paper is a framework for collaborative event recommendation. The active learning method provides a natural extension to this framework. In addition, based on the user study, we generate a new dataset for event prediction.

2. RANKING PROBLEM

We consider a ranking problem, where, given a set of users U and known user feedback on a set of past events E_P, the goal is to generate rankings of a set of *future* events E_F, adapted to each of the users' preferences.

We assume that user feedback is available in the form of *pairwise preferences*. In practice, events compete with each other on user resources, where a small subset of events can be usually attended in a given time frame. For instance, on a particular week, one may be able to attend up to two talks, due to a busy schedule; this means that in deciding which events to attend, the user should compare between alternative events taking place on that week, selecting those that agree most with his or her tastes. In order to model

[1]We will use the words talk and seminar interchangeably.

one's individual ranking function, we therefore consider user inter-item preferences. Specifically, a preference pair is of the form $(e_j, e_k) \in R(u)$, implying that event e_j is preferred to event e_k by user u.

User feedback in the form of preference pairs can be elicited explicitly, by collecting user input on pairs of events: the user indicates which of the events is closer to his or her interests. Alternatively, if it is known that the user has attended (or, has been interested in attending) a subset of the events within a given time frame, then it is reasonable to conclude that those events are preferred by the user over the other events in the same time frame. More formally, we will associate every past event $e_j \in E_P$ with a time stamp t_j, denoting the time in which the event took place. We can split the time span in which past events E_P have occurred into distinct *reference time frames*, where every time stamp t_j maps to a single reference frame T_i. It is assumed that all the events that take place in the same time frame T_i are evaluated against each other. Let $E_P^+(T_i, u)$ denote the subset of events that the user u liked (or attended) in time period T_i, and $E_P^-(T_i, u)$ be the complimentary subset that user u is known not to have liked. Then, this type of feedback can be decomposed into a set of pairs with the expectation that $(e_j, e_k) \in R(u)$ whenever $e_j \in E_P^+(T_i, u)$ and $e_k \in E_P^-(T_i, u)$. Future events are handled similarly, where the stream of incoming events is discretized into reference time frames.

This setting is similar to processing user click logs in information retrieval [11], where the user is provided with a ranked list and clicks on items of interest. While, in the user click setting, users may scan only the top few items of a ranked list, we infer pairwise preference relations exhaustively: because the number of talks each user can pick from is relatively small (see Figure 1), we assume that all the displayed events are observable. In general, if positive feedback corresponds to actual attendance, then the derived preferences may be noisy as schedule constraints may prevent users from attending events they like. Calendar constraints can be potentially detected by dedicated applications. This is beyond the scope of this work.

Since no feedback is available for future events, recommendations must be based on event descriptions. Each event e_j is represented as a vector of m features, $\underline{x}_j$. User feedback in the form of preference pairs $(e_j, e_k) \in R(u)$ means that announcement feature vector $\underline{x}_j$ is preferred over $\underline{x}_k$. Our goal is to assign a real valued evaluation function for each future event $e_j \in E_F$ and user u. Then, given the evaluated scores, we can generate a ranked list of items per user.

2.1 Content-based recommendations

We consider linear ranking functions where each event feature vector $\underline{x}_j$ is mapped to a score $\underline{\theta} \cdot \underline{x}_j$. The goal is to find parameters $\underline{\theta}$ such that the ranking function best captures past feedback from the user. In order to solve this problem for each user u individually, we can apply the RankSVM formulation [11] as follows:

$$\text{minimize } \frac{1}{2}||\underline{\theta}_u||^2 + C\sum_{jk}\xi_{jk} \qquad (1)$$

subject to $\underline{\theta}_u \cdot \underline{x}_j \geq \underline{\theta}_u \cdot \underline{x}_k + 1 - \xi_{jk}$ for all j and k such that $e_j \in E_P^+(T_i, u)$ and $e_k \in E_P^-(T_i, u)$ for some past time frame T_i. The slack variables $\xi_{jk} \geq 0$ turn the ranking constraints into soft constraints, effectively permitting us to make errors on some pairs that are difficult to capture with a linear ranking function. C is a trade-off parameter specifying how strongly we should try to respect all training constraints that can be set through cross-validation.

2.2 Collaborative recommendations

The RankSVM approach described above requires us to estimate an $m-$dimensional parameter vector $\underline{\theta}_u$ separately for each user regardless of potential similarities across users. We hypothesize that most users base their decisions about events on a smaller number of latent "perceptual" features about the events. In order to uncover these latent feature dimensions, we introduce a $k \times m$ parameter matrix V, shared across all users. We use this matrix to map event descriptions $\underline{x}_j$ into a $k-$dimensional subspace $\underline{x}'_j = V\underline{x}_j$, where $k \ll m$. Each user u will subsequently estimate their $k-$dimensional parameter vector $\underline{\theta}'_u$ to work well as part of the transformed event descriptions $\underline{\theta}'^T_u \cdot \underline{x}'_j = \underline{\theta}'^T_u V\underline{x}_j$. Note that users may hold diametrically opposing views about how to use the latent feature dimensions. We merely impose the constraint that they agree on what these dimensions represent.

The number of parameters we need to estimate across users is now considerably smaller. If N denotes the number of users, then we are estimating $N\,k + k\,m$ parameters instead of $N\,m$. The difference is substantial when $k \ll m$. In fact, our method can be seen as imposing a rank k constraint on the collective parameter choices. If we stack $m-$dimensional row vectors $\underline{\theta}^T_u$, $u = 1, \ldots, N$, into a $N \times m$ parameter matrix Θ, then we impose the constraint that this matrix has a rank k decomposition $\Theta = U\,V$, where the rows of $N \times k$ parameter matrix U correspond to new $k-$dimensional user parameters $\underline{\theta}'_u$, $u = 1, \ldots, N$.

The estimation problem can be defined analogously to the RankSVM formulation. The main difference is that the transformation parameters V are estimated across users. More formally, we find U and V that

$$\text{minimize } \frac{1}{2}\|U\|_F^2 + \frac{1}{2}\|V\|_F^2 + C\sum_{ujk}\xi_{ujk} \qquad (2)$$

subject to $[UV\underline{x}_j]_u \geq [UV\underline{x}_k]_u + 1 - \xi_{ujk}$ for all j, k, and user u such that $e_j \in E_P^+(T_i, u)$ and $e_k \in E_P^-(T_i, u)$ for some past time frame T_i. Here $\|\cdot\|_F^2$ denotes the squared Frobenius norm of the matrix (sum of squared entries). From the point of view of each user, for a fixed transformation parameters V, the estimation problem can be solved with RankSVM as before. Similarly, for a fixed U, we could estimate V with RankSVM across users. Our implementation (details omitted) is slightly more efficient than this alternating minimization approach by instead iteratively estimating rank 1 components of $U\,V$ (one column of U, one row of V).

We note that if event descriptions $\underline{x}_j$ are merely binary indicator vectors for event identities, i.e., having exactly one non-zero component, then our collaborative approach reduces to the typical matrix factorization approach to collaborative filtering with the exception that the error is measured in terms of ranking constraints rather than mean squared rating error.

3. USER STUDY

We conducted a user study to collect user preferences on a sequence of scientific talks. Ideally, we are interested in a

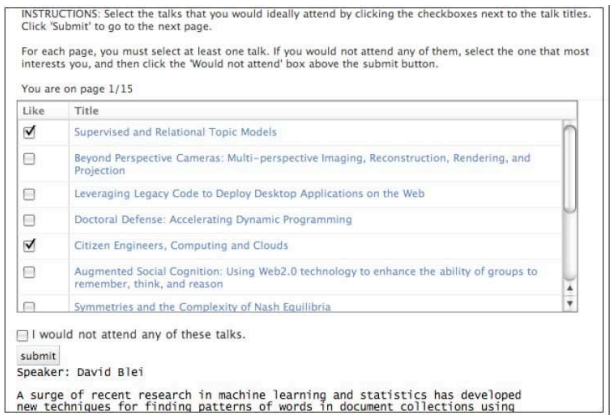

Figure 1: User interface for collecting forced-choice seminar preferences.

scenario where a localization system tracks a person's location indoors [15]; if information about ongoing events (e.g., talks), including event time and room, is maintained, then this type of location-based system can automatically detect a person's presence at events over time. In the current study, however, we elicited *explicit* feedback from users. We examined a realistic scenario: scientific seminars are announced at MIT CSAIL as well as in other institutions via a dedicated email list on a weekly basis. Similarly, in our user study, participants were presented with a list of seminars known to have taken place during a single calendar week. As illustrated in Figure 1, the list contained the talk titles, where the content of each announcement could be displayed by clicking on its title line. An example seminar announcement is displayed in Figure 2. The list of titles for a particular week was ordered randomly so as to remove any overt biases due to display order. The participants were requested to select which of the seminars included in the list they would be interested in attending. In the online form, interest in a seminar was indicated by marking a checkbox next to its title, and the labels for the whole list were delivered by pressing a *submit* button. We required that at least one talk is selected as relevant. In cases where the users felt they were forced to make a choice, i.e., they would prefer not to attend any of the talks, they could indicate this with a checkbox presented at the top of the screen. However, we will assume that their selection still carries information about relative ranking of the alternatives. In order to mimic time flow, and to assure that the reference time frame is weekly, the study participants were not allowed to make changes to previously submitted feedback.

We collected user feedback for two sets of 15 consecutive weeks of seminar announcements using this procedure. Both sets have been originally published on the CSAIL seminar email list, where the first week sequence starts at the first week of September 2007, and the second starts at the sixth week of year 2009. The announcements included all the talks published, where duplicate announcements, as well as any posts that do not correspond to a scientific seminar, were excluded. Thirty CSAIL participants completed the study using the first dataset of seminar announcements, including mostly graduate students and research associates. In addition, 56 graduate students of the computer science school at

```
User Interfaces and Algorithms for Anti-Phishing
HCI Seminar Series Fall 2007
Speaker:  Jason Hong
Speaker Affiliation:  Carnegie Mellon HCI Institute
Host:  Rob Miller
Host Affiliation:  MIT CSAIL

Phishing is a growing plague on the Internet, cost-
ing customers and businesses anywhere between $1-2.8
billion dollars a year.  In this talk, I will present
an overviwe of our work in Supporting Trust Decisions
project.
Our work focuses on developing better user interfaces
to help people make better trust decisions, develop-
ing training mechanisms to teach people not to fall
for phish, and better machine learning and informa-
tion retrieval algorithms than can automatically detect
phishing attacks.

BIO: Jason Hong joined the School of Computer Science
at Carnegie Mellon University...
```

Figure 2: An example email seminar announcement. We drew LDA topic models from seven years worth of similar announcements.

Carnegie Mellon University completed the study, using the second sequence of seminar announcements. Therefore, in both cases, the participants belonged to the target audience of the seminar announcements.

Table 1 shows relevant weekly statistics about the seminar announcements presented and the corresponding user feedback. The frequency of talks varies widely throughout each 15 week period, ranging from 2 to 21 talks on a given week. The table also includes the average number of talks judged as relevant across users per week, and the number of derived preference pairs. Overall, 8.6% of weekly responses were marked as "forced" selections, where the user found no talk as relevant to their interests. Interestingly, participants often selected approximately two talks to attend, regardless of the number shown, mirroring typical real-life attendance.

4. EVENT FEATURES

Each seminar announcement can be viewed as a document (see Figure 2). There are many ways of turning documents into fixed-length feature vectors $\underline{x}_j$. Typically, each coordinate of the feature vector would correspond to a word occurrence such as term-frequency inverse document frequency (TF-IDF) weighted word count [14, 17]. The intuition behind this approach is that topics that users are interested in may be associated with specific terminology and therefore particular coordinates of $\underline{x}_j$. We use TF-IDF weighted word counts as a baseline feature representation.

An alternative and potentially better feature representation can be obtained by identifying topics from the seminar announcements. For example, users may decide whether to attend a seminar based on the degree of overlap between their research areas and the focus of the talk. Such inferences rely on topics rather than individual words. While identifying topics that reflect human judgement is difficult, topic distributions inferred by methods such as Latent Dirichlet Allocation (LDA) [4] may be sufficient for recommendation purposes.

	Week	No. talks	No. relevant	No. pairs
Survey-1	1	8	2.0 (0.8)	11.2 (3.3)
(at MIT,	2	8	1.9 (0.9)	10.9 (3.5)
30 users)	3	7	1.4 (0.7)	7.2 (2.1)
	4	3	1.3 (0.5)	2.0 (0)
	5	20	2.7 (1.7)	43.7 (22.7)
	6	12	1.8 (1.1)	17.0 (7.7)
	7	5	1.1 (0.3)	4.1 (0.5)
	8	12	1.9 (1.0)	18.2 (7.3)
	9	21	2.3 (2.0)	39.7 (26.2)
	10	17	2.4 (1.4)	33.1 (14.3)
	11	7	1.9 (1.0)	8.8 (2.4)
	12	7	1.7 (1.0)	8.0 (2.4)
	13	5	1.2 (0.6)	4.3 (0.7)
	14	21	2.7 (1.8)	45.6 (23.9)
	15	5	1.2 (0.5)	4.4 (0.8)
Survey-2	1	11	2.5 (1.3)	19.5 (7.2)
(at CMU,	2	8	1.4 (0.6)	8.6 (2.8)
56 users)	3	7	1.6 (0.7)	8.2 (2.4)
	4	11	1.4 (0.8)	12.9 (5.1)
	5	11	1.6 (0.8)	14.3 (5.6)
	6	11	1.7 (1.2)	14.5 (6.6)
	7	11	1.8 (1.2)	15.0 (6.5)
	8	2	1.0 (0.1)	1.0 (0.1)
	9	14	2.3 (1.6)	14.9 (12.0)
	10	13	1.7 (1.1)	17.7 (7.9)
	11	11	1.9 (1.2)	15.7 (7.0)
	12	7	1.2 (0.7)	6.4 (1.2)
	13	17	2.0 (1.5)	27.6 (15.1)
	14	12	1.4 (1.0)	14.1 (6.0)
	15	17	2.6 (2.5)	31.2 (15.9)

Table 1: User study statistics using two different sets of 15 consecutive weeks: number of talk announcements per week, average number and standard deviation (in brackets) of talks considered relevant per week, and the corresponding number and standard deviation of derived preference pairs.

LDA is a generative model over documents. The key assumption in the model is that each document is composed of some subset of K possible topics. The words are generated in a manner that reflects the topic composition. Specifically, to generate a document d from this model, we first draw a sample that determines the topics used in the document. The resulting sample is a distribution over topics $P(z|d)$, $z = 1, \ldots, n$. Subsequently, for each word, we first draw a topic from $P(z|d)$ that the word is associated with, and then the actual word from the topic-dependent model $P(w|z)$. Conversely, given a document, and the model parameters, we can infer the overall frequency of each topic used in generating the words in the document. We use these topic usage frequencies as the feature vector. In other words, each coordinate of $\underline{x}_j$ now corresponds to a frequency that a particular topic was used in generating the seminar announcement.

Learning an LDA model involves estimating the parameters that govern the generation of topic compositions as well as the topic-dependent word distributions from a collection of documents. For this purpose, we employ a large corpus, which includes all MIT CSAIL seminar announcement from May 2002 to July 2009. Overall, this reference corpus includes about 5,000 seminar announcements. We use Gibbs sampling method [8] for learning and inference in the LDA model. (The same corpus is used to derive word and inverse document frequencies for the word-based representation described above.)

In addition to the information contained in the seminar announcements, it is possible to consider other relevant information sources. For example, we could search for abstracts of previous publications by the seminar *speaker*. Since the seminar description is limited, the summaries of related publications or talk summaries may better narrow down the topics covered in the talk. Moreover, related documents would also help in case users' interests stem partly from the speaker's expertise or background. Once inferred from relevant abstracts or documents, speaker features can be simply appended to the feature vector.

In general, there are various other representation schemes that could be considered. For example, semi-structured meta-data, specifying details such as the speaker's name and affiliation, or host details (see Figure 2), could be modeled explicitly. One could also model the user's areas of interest, affiliation or social group. In this paper, however, we perform limited feature exploration as our focus is on applying and evaluating the effect of collaborative recommendation. Others may wish to extend the model through these or other methods with their own examination of the talk and user feedback dataset.

5. EVALUATION

As mentioned above, the user study was designed to mirror a realistic setting, where user responses to ongoing events, whether collected explicitly or implicitly by sensors, is accumulated over time. There are several questions that we are interested in addressing in our evaluation of the user study:

- A primary goal of the evaluation is to examine the conjecture that the proposed collaborative LowRank method can improve prediction performance of an event recommendation system. Towards this end, we compare LowRank against the RankSVM method [11], where a separate model is learned for each individual user.

- We consider several feature sets and information sources, evaluating their contribution to performance, and their interaction with the learning methods evaluated.

- We are interested in estimating the "learning curve" of a seminar events' recommendation system: as the pool of user feedback increases, what is the level of performance one can expect, and its rate of change over time?

- Finally, another part of our evaluation concerns the variance in performance of the recommendation system across the individual users.

In the experiments, we divide the data collected into *train* and a *test* sets. The train set includes the first 10 weeks for which we obtained user feedback (in both user surveys). This labeled data is used as input to the prediction models, where we simulate varying lengths of the learning period by considering increasing number of weeks for which user feedback is available. The testing set includes the seminar announcements for weeks 11-15 (in both surveys), representing future events. In testing, we apply the models learned to generate a ranked list for every user and for each week in the test set. The results are then evaluated against the user feedback collected. Tuning of the learning methods' parameters is performed using the train set.

All the methods applied generate a ranked list of event entities. We evaluate performance in terms of Mean Average Precision (MAP), which is a widely accepted evaluation measure in Information Retrieval [13]. To define MAP, we

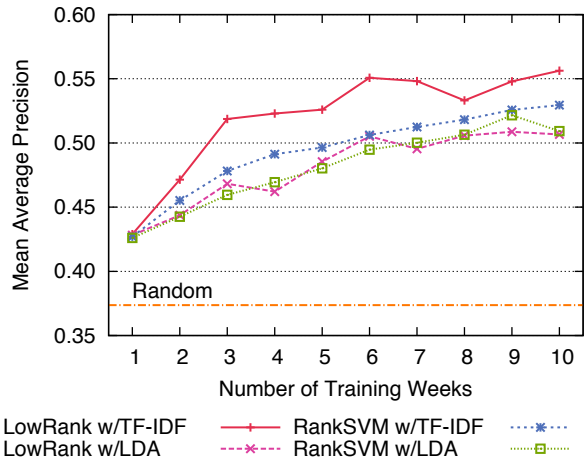

Figure 3: A comparison of RankSVM and LowRank with increasing training data. MAP is averaged across all users for all testing weeks.

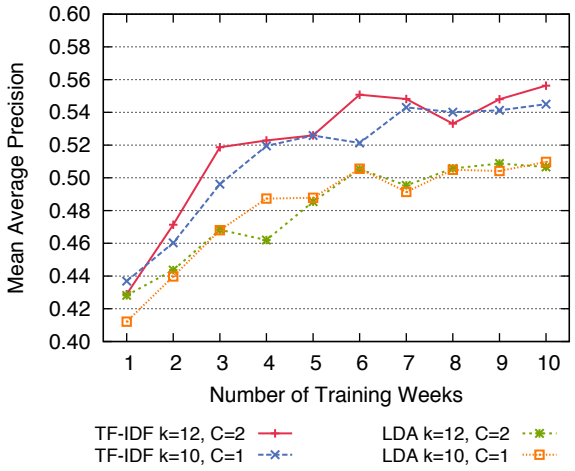

Figure 4: Performance of LowRank with increasing training data using two sets of parameter choices. MAP is averaged across all users for all testing weeks.

first define the *precision at rank* k, $prec(k)$, to be the number of correct entries up to rank k, divided by k—i.e., the precision of the list up to rank k. The *non-interpolated average precision* of the ranking is the average of $prec(k)$ for each position k_i that holds a correct entry:

$$AveragePrecision = \frac{1}{n}\sum_{i=1}^{n} prec(k_i)$$

For example, consider a ranked list of items, where the items at ranks 1,2,5 are known to be correct answers, and those at ranks 3,4 are not; the non-interpolated average precision of this ranked list is $(1 + 1 + 0.6)/3 = 0.87$. The Mean Average Precision (MAP) is the average of the non-interpolated precision scores, over multiple rankings (i.e., over multiple queries).

The MAP measure is strongly correlated with query difficulty, reflected by the length of the ranked list and the size of the correct item set. For example, a list of two items overall, including one correct item, will have a MAP of 0.5 in the worst case; ranking a list of N items with one correct item, however, may reach a lower MAP of $1/N$, etc. For this reason, we evaluate the various methods in terms of MAP on the same set of test queries.

In our knowledge, this work is the first to present and evaluate event prediction via a user study.[2]

5.1 Accuracy vs. amount of feedback

We first evaluate the performance of learning individual models per user using RankSVM , considering several event representation schemes. Specifically, an event is represented as a topic distribution, where the number of topics K is set to 100.[3] An alternative representation considered is a TF-IDF weighted word vector. In our experiments, word inverse document frequency (IDF) counts were obtained from

[2]The underlying dataset will be made available on the first author's homepage.

[3]In tuning experiments, we explored setting the number of latent topics to $K = 10, 25, 50, 100, 200$. While $K = 10$ led to inferior prediction results, the other value selections were found to be comparable.

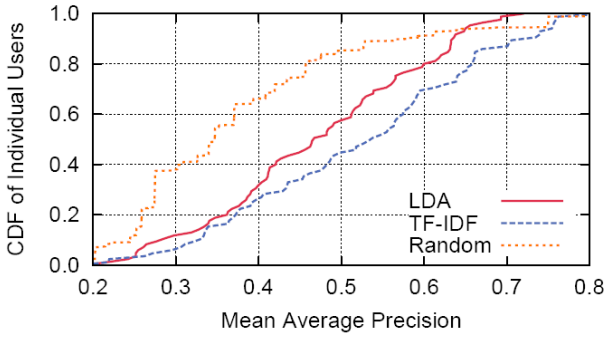

Figure 5: LowRank with three weeks of training as compared to random seminar recommendations. Data shows Mean Average Precision for the full five testing weeks and individual users.

the full corpus of seminar announcements available (Section 4). We apply simple feature selection, where words that appear fewer than three times in the corpus are eliminated. Overall, this results in roughly 10,000 word features. Tong and Koller [17] have previously found a similar word-based representation to be effective for text classification tasks.

One of the goals of our evaluation is to gauge the impact of train set size on performance. We therefore conduct a set of experiments, where an increasing number of weeks (out of the train set available to us) are used for training. Specifically, we train recommendation models based on n weeks worth of data per user, where n ranges from one week of labeled feedback per user, to the full ten weeks worth of feedback per user available in the full train set.

In a real recommendation system, the set of users is dynamic; in particular, users join the recommendation service at different points in time. It is also possible that only a subset of the existing users provide feedback during any given period (a user may be inactive. e.g., away). In the experiments, we therefore allow the known feedback per user to vary in time: if it is assumed that $n < 10$ weeks worth of

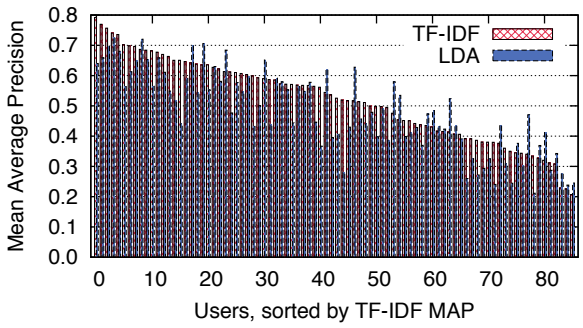

Figure 6: Each pair in the histogram shows a user's MAP for using LowRank with both TF-IDF and LDA feature sets. The data show a strong correlation between predictability using either feature set.

feedback are available per user, then n weeks are selected randomly for each user out of the ten weeks included in the train set. For every value of n, we train recommendation models based on a subset of the training set constructed in the described fashion. In order to eliminate sampling bias, every experiment is repeated 10 times overall. The learned models are then evaluated using the fixed test set, where we report average performance across the 10 trials.

Figure 3 shows global MAP performance over the whole test set, averaged across all users and across trials, for RankSVM using TFIDF and topic feature vectors. The figure displays the results for increasing volume of feedback, starting at one week's worth of user feedback and up to considering feedback for the full ten weeks available. As a naive baseline, the figure includes the results of *random* ordering of the talks in the test set.

There are several trends observed in the results. First, learning is effective, as it is consistently yields results superior to random ordering. As one may expect, learning performance improves over time, as more user feedback on past events becomes available. However, the improvement ratio is relatively modest. We conjecture that new additional topics are introduced in the seminars included in the test set, where this limits the performance of the content-based models that are learned based on past experience.

With respect to the feature scheme, we find that applying RankSVM using the topic and TFIDF representation vectors yields comparable performance. Elsewhere, small improvements have been obtained using topic features on the task of text classification [1].

5.2 Collaborative prediction vs baseline

Figure 3 shows the results of applying the LowRank collaborative approximation method for increasing volume of training data, using TFIDF and topic feature representations. We fixed the regularization term to $C = 2$, and the rank of the parameter matrix V to $k = 12$. As shown, LowRank using TFIDF features yields superior performance across the full range of inspected train set size. In addition, the learning curve in this case is steeper, where performance given three weeks of training data exceeds the best performance observed with the RankSVM models using eight weeks worth of training data. A shorter learning time translates to an improved user experience, and overall better system utility, as new users join the system over time.

Applying LowRank using the topic features gives comparable performance to RankSVM . LowRank gives preferable results to RankSVM if TFIDF vectors are used. Since the TFIDF representation is much more sparse, LowRank has more direct control in this case over the useful subspace to use.

We found that a matrix rank of at least $k = 8$ is needed to reach good performance in the experiments. A small variance was observed for higher values of k. Figure 4 shows the performance curve of LowRank setting the rank to $k = 10, 12$ and the regularization term to $C = 1, 2$. As shown, the sensitivity to parameter values in this range is small.

5.3 Individual users' performance

While results have been discussed so far in a global fashion, a question of interest is what variance can be expected among individual users' performance. In the experiments, we found this variance to be relatively large. Figure 5 presents the results of RankSVM using topic and TFIDF feature sets and (10 samples of) three weeks worth of user feedback for training. The figure shows the cumulative rate of MAP performance for an individual user for each level of MAP performance observed. According to the figure, for about 50% of the users, average MAP across runs was 0.55 or higher using LowRank with TFIDF features. If topic features are used, average MAP performance for about 50% of the users is 0.47 or more. Random recommendation yields lower MAP of 0.35 and above for about 50% of users. As mentioned earlier, MAP can vary between users due to variance of the relevant item set size (Table 1); in general, accuracy is expected to be higher for users who are interested in a larger number of talks. Similarly, differences in performance between users may correspond to the extent to which the user is interested in general areas, versus specialized sub-topics. In the latter case, data scarcity is expected to be more pronounced. Predictions for a user who has wide interest areas are also likely to include more relevant items on average than for a user whose interest areas are narrow.

Finally, Figure 6 gives another view of the results using LowRank with topic and TFIDF features on the full test set, for individual users. In addition to detailing performance level per user, this figure shows a strong correlation in individual user performance between the models trained with the different feature sets.

6. EXTENSIONS: ACTIVE ELICITATION

There are two problems we hope to remedy by actively eliciting feedback from users. The first problem is general data sparsity. In a working system, we expect to highlight only about 5% of all the announcements that the user receives (say, in the course of one month). The goal is therefore to properly rank the top 5% amongst themselves as well as separate this set from the remaining announcements. However, realistically, a single user is unlikely to provide explicit feedback for more than a few weeks worth of announcements. As a result, many of their top choices may not appear in the training data. In order to avoid this problem, it seems necessary to quickly focus the announcements presented to the user to be mostly in the top 5% of their ranking. For example, based on initial information from the user, an active selection approach could present to the user some predicted

top 5% announcements from the overall pool of announcements, some 5-10% percentile announcements so as to learn to separate them from the top, and some randomly chosen announcements (or according to the normal weekly schedule) in order to avoid missing major areas. The quality of these predicted rankings, and therefore queries themselves, would improve with the user feedback.

The second and related issue is coverage across areas of interests. In part, this is achieved by adopting the collaborative formulation described earlier. We only need to identify user interests by selecting a parameter vector (direction) in the low dimensional (shared) perceptual coordinate space. The perceptual feature coordinates can be thought as representing weighted subsets of topics and therefore resist specifying user interests too narrowly.

Consider a fixed mapping V to the lower dimensional perceptual space. This mapping, shared across users, can be estimated robustly based on limited initial feedback from users. For each user u, we represent the uncertainty in their parameters $\underline{\theta}'_u$ with a distribution $P(\underline{\theta}'_u)$, a discrete distribution over a set of alternative parameters. This distribution is updated based on user feedback. We assume that the user responds to a pair of announcements in a manner that reflects some $\underline{\theta}'_u$ but also allowing flipping noise. The score for a candidate pair $(\underline{x}_i, \underline{x}_j)$ is evaluated as follows. Let $y_{ij} \in \{-1, 1\}$ represent the user response if they are presented with $(\underline{x}_i, \underline{x}_j)$. We model $P(y_{ij}|\underline{x}_i, \underline{x}_j; \underline{\theta}'_u)$ as follows:

$$P(y_{ij} = 1|\underline{x}_i, \underline{x}_j; \underline{\theta}'_u) = \begin{cases} (1-\epsilon), & \text{if } \underline{\theta}'_u \cdot (V\underline{x}_i) > \underline{\theta}'_u \cdot (V\underline{x}_j) \\ \epsilon, & \text{otherwise} \end{cases}$$

where ϵ is a small flipping probability. The expected response, $P(y_{ij}|\underline{x}_i, \underline{x}_j)$, is obtained by averaging relative to the current belief about $\underline{\theta}'_u$, i.e., with respect to $P(\underline{\theta}'_u)$. The score for a candidate pair of announcements to be presented to the user for feedback is then the expected information gain: the entropy of the parameter choices prior to feedback, $-\int P(\underline{\theta}'_u) \log P(\underline{\theta}'_u) d\underline{\theta}'_u$, minus the expected entropy after the feedback

$$-\sum_{y_{ij}} \int P(\underline{\theta}'_u) P(y_{ij}|\underline{x}_i, \underline{x}_j; \underline{\theta}'_u) \log \frac{P(y_{ij}|\underline{x}_i, \underline{x}_j; \underline{\theta}'_u) P(\underline{\theta}'_u)}{P(y_{ij}|\underline{x}_i, \underline{x}_j)} d\underline{\theta}'_u$$

The integrals are tractable since, by assumption, $P(\underline{\theta}'_u)$ has mass only on a set of discrete alternatives over the $k-$ dimensional unit ball (predictions are invariant to scale). The active learning method successively selects pairs with the highest information gain, updating $P(\underline{\theta}'_u)$ based on each response.

In order to evaluate the active learning approach in an accurate fashion, we would like to conduct another user study, where users are asked to provide pairwise feedback for preference pairs, selected based on their previous feedback, in an online procedure. So far, we have conducted preliminary experiments in a simulated mode, based on feedback already acquired. As before, we used the first 10 weeks of seminar announcements as possible data for training. The active learning method was initialized with 2-4 weeks of data using the collaborative approach. The purpose of this step is to identify the initial shared coordinate dimensions. Subsequently, we selected pairs of announcements from the training data, separately for each user. The selections are restricted to pairs that are informative (user selected one but not the other). The selection of pairs was based on reducing uncertainty about the $k-$dimensional user parameters $\underline{\theta}'_u$.

We do not report full results in this paper. As a single example, we found that if the data of the first three weeks is considered, then three active learning pairs give better performance than training on full data individually in some cases. These are encouraging results, and we intend to investigate them further.

7. RELATED WORK

The main focus of this work is on the general problem of event recommendation. The user study conducted, however, concerns the specific application of event recommendation to scientific talks. Previously, researchers have considered a related task of automatically recommending scientific papers to reviewing committee members. In most works, this problem has been approached using a content-based methods [2, 7, 18]. Dumais and Nielsen [7] compute paper-reviewer similarity, based on paper abstracts and titles on one hand and relevant abstracts supplied by the reviewers on the other hand, using latent semantic indexing (LSI). Their results are up to 40% better than a random baseline in terms of accuracy at rank 10. Interestingly, they indicate that reviewers are unable to judge their own interests with perfect consistency; also, the performance of the automated system was found similar to that of human judges. Basu et-al [2] evaluate the contribution of various information sources to the task of recommending papers to reviewers. They use WHIRL, a query language that accommodates similarity metrics. In addition, they experiment with collaborative filtering, where feedback is fed by reviewers in an online fashion, and recommendations are generated based on the feedback stream using methods like KNN. According to their results, content-based approaches yield better results than pure collaborative filtering on this task. Another work [5] considers collaborative filtering given reviewer 'bids', expressing interest or disinterest of reviewers in specific papers, as available feedback. While these previous works apply collaborative filtering in "traditional" settings, based on common pools of items and users, we study a different problem, where the sets of known items and the items for which recommendation takes place are distinct.

The problem of collaborative filtering for event recommendation has not received much attention, in our knowledge. A recent work proposes a hybrid content and collaborative filtering approach for event recommendation, within a fuzzy relational framework [6]. Their rationale is similar to ours, where the underlying goal is recommending future events if they are similar to past events that similar users have liked. The approach proposed is not evaluated empirically. In contrast, in this paper we extend a more popular recommendation framework to fit the settings of event recommendation; we also present a relevant user study, and make it available for the research community.

The LowRank method suggested in this paper takes a matrix factorization approach to collaborative recommendation [12]. There is vast literature about collaborative filtering via low rank approximation (e.g., [9, 10]).

8. CONCLUSION

We discussed the problem of recommending items for which no previous feedback exists, focusing on the problem of fu-

ture event recommendation. We introduced a low rank collaborative approach in this setting. A user study was conducted to simulate a recommendation of scientific seminars on a weekly basis. Our empirical results based on this user study show that the proposed collaborative method outperforms content-based recommendation on this problem. Since the collaborative method approach uses explicit feedback, we proposed an active learning extension to the approach that is aimed at reducing the amount of explicit feedback required from the user. We plan to evaluate this method using a dedicated user study in the future. In addition, we believe that as outdoor and indoor localization systems evolve, it will become possible to track users' interests in items such as events directly by detecting their attendance at known events.

9. ACKNOWLEDGMENTS

This research is funded by Nokia Research.

10. REFERENCES

[1] S. Banerjee. Improving Text Classification Accuracy using Topic Modeling over an Additional Corpus. In *Proceedings of the International ACM SIGIR Conference*, 2008.

[2] C. Basu, H. Hirsh, W. W. Cohen, and C. Nevill-Manning. Technical paper recommendation: A study in combining multiple information sources. *Journal of Artificial Intelligence Research*, 1, 2001.

[3] J. Bennett and S. Lanning. The Netflix Prize. In *KDD Cup and Workshop*, 2007.

[4] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[5] D. Conry, Y. Koren, and N. Ramakrishnan. Recommender systems for the conference paper assignment problem. In *Proceedings of the third ACM conference on Recommender systems*, 2009.

[6] C. Cornelis, X. Guo, J. Lu, and G. Zhang. A fuzzy relational approach to event recommendation. In *Proceedings of the Indian International Conference on Artificial Intelligence*, 2005.

[7] S. T. Dumais and J. Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *Proceedings of the International ACM SIGIR Conference*, 1992.

[8] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 11(1), 2004.

[9] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proc. of the Conference on Research and Development in Informaion Retrieval (SIGIR)*, pages 259–266, Aug. 2003.

[10] T. Hofmann. Latent Semantic Models for Collaborative Filtering. *ACM Trans. Inf. Syst.*, 22(1), 2004.

[11] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

[12] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8), 2009.

[13] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[14] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the International Conference on Digital Libraries*, 2000.

[15] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an Organic Indoor Location System. In *Proc. of the International Conference of Mobile Systems, Applications, and Services (MobiSys)*, 2010.

[16] G. Strang. *Linear Algebra and its Applications*. Harcourt Brace Jovanovich College Publishers, Orlando, FL, 3rd edition, 1988.

[17] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2, 2002.

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