# Leveraging User Libraries to Bootstrap Collaborative Filtering

Laurent Charlin*
Princeton University
Dept of Computer Science
Princeton, NJ, USA
lcharlin@cs.princeton.edu

Richard S. Zemel
University of Toronto
Dept of Computer Science
Toronto, ON, Canada
zemel@cs.toronto.edu

Hugo Larochelle
Université de Sherbrooke
Département d'informatique
Sherbrooke, QC, Canada
hugo.larochelle@usherbrooke.ca

## ABSTRACT

We introduce a novel graphical model, the collaborative score topic model (CSTM), for personal recommendations of textual documents. CSTM's chief novelty lies in its learned model of individual libraries, or sets of documents, associated with each user. Overall, CSTM is a joint directed probabilistic model of user-item scores (ratings), and the textual side information in the user libraries and the items. Creating a generative description of scores and the text allows CSTM to perform well in a wide variety of data regimes, smoothly combining the side information with observed ratings as the number of ratings available for a given user ranges from none to many. Experiments on real-world datasets demonstrate CSTM's performance. We further demonstrate its utility in an application for personal recommendations of posters which we deployed at the NIPS 2013 conference.

## 1. INTRODUCTION

With the advent of online services and the wealth of information made accessible through them systems with the ability to filter the relevant from the irrelevant, such as recommendation systems, are becoming ubiquitous. Collaborative filtering (CF) models have rapidly established themselves as the *de facto* standard for many recommendation tasks where user-item preferences, scores or ratings, are available. In addition to such preferences, *side information* about users or items, for example other information that may be collected by an online service, is often available. Side-information has been particularly useful to address the *cold-start* problem that plagues collaborative filtering systems. Cold-start refers to a regime where scores for a set of users or items are unavailable or scarce. In some cases side information has also been shown to improve on the performance of collaborative filtering models in non-cold start (*warm-start*) data regimes [7, 20].

In this paper we are interested in the task of document recommendation using both user-item preferences and side information. The primary novelty of our work lies in leveraging a particular form of side information: the content of documents associated with users, which we call *user libraries*. A typical scenario that can be modelled in this way is scientific-paper recommendation for researchers; for example, Google Scholar recommends papers based on an individual's profile. A second scenario is paper-reviewer assignment, where each reviewer's previously published papers can be used to assess the match between their expertise and each submitted paper. Another relevant application domain is book recommendation, as online book merchants typically allow users to collect items in a virtual container akin to a personal library.[1] In each case a user's library, or side information, consists of documents which are not necessarily explicitly rated but nonetheless likely contain information about a user's preferences.

To model user-item scores as well as user and item content we introduce a novel directed graphical model. This model uses twin topic models, with shared topics, to model the side information. User and item topic proportions are then used as features to predict user-item scores with a collaborative filtering model. The collaborative filtering component allows the model to effectively make use of the side information with varying amounts of observed scores. We demonstrate empirically that the model outperforms several others on three datasets in both cold and warm-start data regimes. We further show that the model automatically learns to gradually trade off the use of side information in favor of information learned from user-item scores as the amount of user preference-data increases.

## 2. MODEL

Our approach to document recommendation relies on having: a) a set of observed user-item preferences ($\{r_{ud}\}$); b) contents of the items ($\{\mathbf{w}_d^s\}$); and c) the content of user-libraries ($\{\mathbf{w}_u^a\}$). The model's aim is to utilize the content in its user-item score predictions (which can then be used to recommend items to users). This contrasts with standard CF, which is not content-based.

Our content-based model is mediated by topics: we learn a shared topic model from the words of the documents and user libraries. We represent topic proportions with a normal

*Work done primarily while author was at the University of Toronto.


*KDD'14,* August 24–27, 2014, New York, NY, USA. 
ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623663.

[1]For example., Amazon's Kindle and Kobo's tablets have an option for users to populate their libraries, while Barnes and Nobles' Nook gives users an *active shelf*.

distribution and realized topics $\mathbf{z}^u$ and $\mathbf{z}^d$ using the logistic normal [5]. User and item topic proportions offer a compact representation of user and item side information. We use these representations as covariates in a regression model to predict user-item preferences. The regression has two sets of parameters. The first are user-specific parameters on the item-topics covariates ($\boldsymbol{\gamma}_u$). The second are *compatibility parameters*, which are shared across users and items, and are based on the compatibility between the item topics and the topics of the user library ($\boldsymbol{\theta}$).

We now introduce the complete graphical model for this *collaborative score topic model (CSTM)*. A graphical representation of the model is given in Figure 1. The associated generative model is:

- Draw compatibility parameters: $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \lambda_\theta I)$
- Draw shared-user parameters: $\boldsymbol{\gamma}_0 \sim \mathcal{N}(\mathbf{0}, \lambda_{\gamma_0} I)$
- For each user $u = 1 \dots U$:
  - Draw individual-user parameters: $\boldsymbol{\gamma}_u \sim \mathcal{N}(\mathbf{0}, \lambda_\gamma I)$
  - Draw user-topic proportions: $\mathbf{a}_u \sim \mathcal{N}(\mathbf{0}, \lambda_a I)$
- For each document $d = 1 \dots D$:
  - Draw document-topic proportions: $\mathbf{s}_d \sim \mathcal{N}(\mathbf{0}, \lambda_s I)$
- For all of user $u$'s user-library words, $n = 1 \dots N$ :[2]
  - Draw $z^a_{un} \sim \text{Mult}(\text{softmax}(\mathbf{a}_u))$
  - Draw $w^a_{un} \sim \text{Mult}(\beta_{z^a_{un}})$
- Repeat the above for all of document $d$'s $M$ words
- For each user-document pair $(u, d)$, draw scores: $r_{ud} \sim \mathcal{N}((\mathbf{a}_u \otimes \mathbf{s}_d)^T \boldsymbol{\theta} + \mathbf{s}_d^T(\boldsymbol{\gamma}_0 + \boldsymbol{\gamma}_u), \sigma)$

where $\mathcal{N}(\mu, \sigma^2)$ represents a normal distribution with mean $\mu$ and variance $\sigma^2$, $\otimes$ stands for the elementwise product, $\text{softmax}(\mathbf{v}) = \frac{\exp(\mathbf{v})}{\sum_{k'} \exp(v_{k'})}$, and $I$ is the identity matrix.

The specific parametrization of the preference regression is important. Our model is designed to perform well in both cold-start and warm-start data regimes. In cold-start settings the model needs the user's side information to predict user-item preferences. When the amount of observed preferences increases the model can gradually leverage that information, smoothly combining it with information gleaned from the side information to refine its model of missing preferences. To accomplish this, the regression model is separated in two parts: one part that exploits user side information ($(\mathbf{a}_u \otimes \mathbf{s}_d)^T \theta$) and another that does not ($\mathbf{s}_d^T(\boldsymbol{\gamma}_0 + \boldsymbol{\gamma}_u)$).

Item side information is incorporated by modulating the user information through an element-wise product. The weights $\boldsymbol{\theta}$ then serve several purposes: 1) they can act to amplify or reduce the effect of certain topics (for example diminish the influence of topics bearing little preference information); 2) they allow the model to more easily calibrate its output to the range of observed preference values; 3) changing the magnitude of $\boldsymbol{\theta}$ allows the model to control how much it uses the side information for preference prediction.

When user-item preferences are more abundant, the model can use them to learn a user-specific model, $\boldsymbol{\gamma}_u$, over item features. Note that these user-specific parameters are combined with a shared set of parameters, $\boldsymbol{\gamma}_0$, which allows for some transfer across users. The individual's $\boldsymbol{\gamma}_u$ can be used to increase that user's reliance on user-item preferences at the possible expense of item side information, as the joint magnitude of the $\boldsymbol{\gamma}$'s defines the weights associated with this part of the model.

[2]For simplicity, we'll assume in the notation that all user-libraries contain $N$ words and all item documents contains $M$ words.

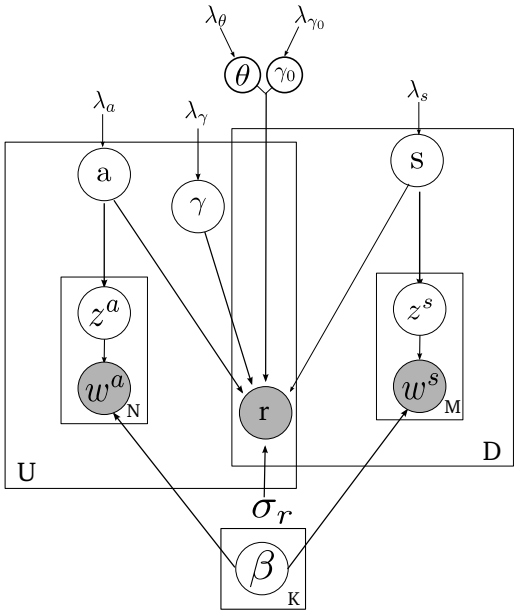


**Figure 1: Graphical model for CSTM.**

Our model learns a single set of topics to model user and item content. Sharing topics ensures that the user and item representations ($\mathbf{a}_u$ and $\mathbf{s}_d$ $\forall u, \forall d$) are aligned and render their element-wise product meaningful.

## 2.1 The relationship between CSTM and standard models

Simplifying the proposed CSTM model in various ways produces other models that have been used for similar tasks. First, setting $\boldsymbol{\gamma}_0$ and $\boldsymbol{\gamma}_u$ $\forall u$ to zero and $\boldsymbol{\theta}$'s to a vector of ones we obtain a version of a language model (LM) first used by [12] for a similar task.[3] [12] and [7] used the LM in a preference prediction task and found its performance particularly strong in low-data regimes.

Further, setting $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}_0$ to zero we obtain an individual user regression model (LR), which was shown to outperform purely collaborative filtering models in a similar preference prediction task with textual side information [7].

By modelling preferences as a combination of user features and item features, our model can also be seen as an instance of collaborative filtering. Collaborative filtering models have proven to be extremely powerful for missing preference prediction problems [15, 14, 3].

Finally, we have opted to represent topic proportions using a normal distribution instead of the more standard simplex representation used in latent Dirichlet allocation (LDA) [6]. This parametrization was proposed for correlated topic mod-

[3]One difference with our model is that since we represent users and items in topic space we do not have to handle normalization nor smoothing issues which are typical of word-space models.

els [5]; in our case we utilize the logistic normal due to its representational form, and not as a means of learning topic correlations.[4] Compared to a multinomial the normal distribution adds a level of flexibility that may be useful to better calibrate CSTM's preference predictions; the drawback is additional complexity in model inference.

# 3. LEARNING AND INFERENCE

For learning we use a version of the EM algorithm where we alternate between updates of the user-item specific variables ($\mathcal{H} = \{\{\boldsymbol{\gamma}_u\}, \{\mathbf{a}_u\}, \{\mathbf{s}_d\}, \{\mathbf{z}^a\}, \{\mathbf{z}^s\}\}$) in the E-step and updates of the parameters or shared variables ($\Theta = \{\boldsymbol{\gamma}_0, \boldsymbol{\theta}, \boldsymbol{\beta}\}$) in the M-step. The inference and learning procedures are similar to those proposed for nonconjugate LDA models in [20]. The general EM algorithm is shown in Algorithm 1.

## E-Step

Inference in this model being intractable, we must rely on approximations when manipulating the posterior over the user-item specific varaibles. The log-posterior over user-item variables, given the fixed model parameters and the data, is

$$\begin{aligned}\mathcal{L} :=& -\frac{1}{2\lambda_a^2}\sum_u^U \mathbf{a}_u^T\mathbf{a}_u - \frac{1}{2\lambda_s^2}\sum_d^D \mathbf{s}_d^T\mathbf{s}_d - \frac{1}{2\lambda_\gamma^2}\sum_u^U \boldsymbol{\gamma}_u^T\boldsymbol{\gamma}_u \\ &- \frac{1}{2\sigma_r^2}\sum_{(u,d)\in\mathcal{O}}\Big(r_{ud} - ((\mathbf{a}_u \otimes \mathbf{s}_d)^T\boldsymbol{\theta} + (\boldsymbol{\gamma}_0 + \boldsymbol{\gamma}_u)^T\mathbf{s}_d)\Big)^2 \\ &+ \sum_{u,n}^{U,N} \log\frac{\exp(a_{uz^a_{un}})}{\sum_j \exp(a_{uj})} + \sum_{d,m}^{D,M}\log\frac{\exp(s_{dz^s_{dm}})}{\sum_j\exp(s_{dj})} \\ &+ \sum_{u,n}^{U,N}\log\beta_{z^a_{un},w^a_{un}} + \sum_{d,m}^{D,M}\log\beta_{z^s_{dm},w^s_{dm}} - \log Z(\Theta) \qquad (1)\end{aligned}$$

where $\mathcal{O}$ stands for the set of observed preferences and $Z(\Theta)$ is an intractable normalizing constant of the posterior, in part because $\mathbf{a}_u$ and $\mathbf{s}_d$ cannot be analytically integrated out since they are not conjugate to the distribution over topic assignments [5].

We address this computational issue by employing variational approximate inference [11]. For the topic-proportion and regression variables $\{\mathbf{a}_u\}, \{\mathbf{s}_d\}, \{\boldsymbol{\gamma}_u\}$, we use a Dirac delta posterior parameterized by its mode $\{\widehat{\mathbf{a}}_u\}, \{\widehat{\mathbf{s}}_d\}, \{\widehat{\boldsymbol{\gamma}}_u\}$. For the topic-assignment variables $\{\mathbf{z}^a\}, \{\mathbf{z}^s\}$, we instead utilize a mean-field posterior. The full approximate posterior is thus:

$$\begin{aligned}q(\{\mathbf{a}_u\}, \{\mathbf{s}_d\}, \{\boldsymbol{\gamma}_u\}, \{\mathbf{z}^a_u\}, \{\mathbf{z}^s_d\}) =& \left(\prod_u^U \delta_{\widehat{\boldsymbol{\gamma}}_u}(\boldsymbol{\gamma}_u)\right) \\ &\left(\prod_u^U \delta_{\widehat{\mathbf{a}}_u}(\mathbf{a}_u)\prod_n^N \phi^a_{unz^a_u}\right)\left(\prod_d^D \delta_{\widehat{\mathbf{s}}_d}(\mathbf{s}_d)\prod_m^M \phi^s_{dmz^s_d}\right)\end{aligned}$$

where $\delta_\mu(x)$ is the delta function with mode $\mu$ and $\{\boldsymbol{\phi}^a_u\}$,$\{\boldsymbol{\phi}^s_d\}$ are the mean-field parameters (e.g., $\boldsymbol{\phi}^a_u$ is a matrix whose entries $\phi^a_{unj}$ are the probabilities that the $n^{\text{th}}$ word in user $u$'s library belongs to topic $j$).

[4]Since learning topic correlations has been found to improve on standard LDA, it is possible that learning the topic correlations could also improve our model.

Approximate inference entails finding the variational parameters $\{\widehat{\mathbf{a}}_u\}, \{\widehat{\mathbf{s}}_d\}, \{\widehat{\boldsymbol{\gamma}}_u\}, \{\boldsymbol{\phi}^a_u\}, \{\boldsymbol{\phi}^s_d\}$ that minimize the KL-divergence with the true posterior

$$\begin{aligned}&\mathrm{KL} := -\mathrm{E}_q[\mathcal{L}] - \mathrm{H}(q) \\ &= \frac{1}{2\lambda_a^2}\sum_u^U \widehat{\mathbf{a}}_u^T\widehat{\mathbf{a}}_u + \frac{1}{2\lambda_s^2}\sum_d^D \widehat{\mathbf{s}}_d^T\widehat{\mathbf{s}}_d + \frac{1}{2\lambda_\gamma^2}\sum_u^U \widehat{\boldsymbol{\gamma}}_u^T\widehat{\boldsymbol{\gamma}}_u \\ &+ \frac{1}{2\sigma_r^2}\sum_{(u,d)\in\mathcal{O}}\Big(r_{ud} - ((\widehat{\mathbf{a}}_u \otimes \widehat{\mathbf{s}}_d)^T\boldsymbol{\theta} + (\boldsymbol{\gamma}_0 + \widehat{\boldsymbol{\gamma}}_u)^T\widehat{\mathbf{s}}_d)\Big)^2 \\ &- \sum_{u,n,k}^{U,N,K}\phi^a_{unk}\left(\log\frac{\exp(\widehat{a}_{uk})}{\sum_j\exp(\widehat{a}_{uj})} + \log\beta_{k,w^a_{un}} - \log\phi^a_{unk}\right) \\ &- \sum_{d,m,k}^{D,M,K}\phi^s_{dmk}\left(\log\frac{\exp(\widehat{s}_{dk})}{\sum_j\exp(\widehat{s}_{dj})} + \log\beta_{k,w^s_{dm}} - \log\phi^s_{dmk}\right) \\ &+ \text{constant.} \qquad (2)\end{aligned}$$

Our strategy is to perform one pass of coordinate descent, optimizing each set of variational parameters given the others.[5]

For $\widehat{\boldsymbol{\gamma}}_u$, we obtain a closed-form update by differentiating the above equation and setting the result to 0:

$$\begin{aligned}\widehat{\boldsymbol{\gamma}}_u =& \frac{1}{\sigma_r}\sum_{d\in\mathcal{O}(u)}(r_{ud} - (\widehat{\mathbf{s}}_d \otimes \widehat{\mathbf{a}}_u)^T\boldsymbol{\theta} - \widehat{\mathbf{s}}_d^T\boldsymbol{\gamma}_0)\widehat{\mathbf{s}}_d^T \\ &\left(\sum_{d\in\mathcal{O}(u)}\widehat{\mathbf{s}}_d\widehat{\mathbf{s}}_d^T/\sigma_r + \lambda_\gamma I/2\right)^{-1}\end{aligned}$$

where $\mathcal{O}(u)$ is the set of indices for documents that user $u$ has rated. The $\{\widehat{\mathbf{a}}_u\}, \{\widehat{\mathbf{s}}_d\}$ parameters do not have closed-form solutions, hence we resort to optimization using conjugate gradient descent. We report the derivatives with respect to the posterior KL:

$$\begin{aligned}\frac{\partial\mathrm{KL}}{\partial\widehat{\mathbf{a}}_u} =& \lambda_a\mathbf{a}_u - \frac{1}{\sigma_r}\sum_{d\in\mathcal{O}(u)}(r_{ud} - \hat{r}_{ud})(\widehat{\mathbf{s}}_d \otimes \boldsymbol{\theta}) \\ &+ N\frac{\exp(\widehat{\mathbf{a}}_u)}{\sum_j\exp(\widehat{a}_{uj})} - \sum_n \boldsymbol{\phi}^a_{un} \\ \frac{\partial\mathrm{KL}}{\partial\widehat{\mathbf{s}}_d} =& \lambda_s s_d - \frac{1}{\sigma_r}\sum_{u\in\mathcal{O}(d)}(r_{ud} - \hat{r}_{ud})(\widehat{\mathbf{a}}_u \otimes \boldsymbol{\theta} + \boldsymbol{\gamma}_0 + \widehat{\boldsymbol{\gamma}}_u) \\ &+ M\frac{\exp(\widehat{\mathbf{s}}_d)}{\sum_j\exp(\widehat{s}_{dj})} - \sum_n \boldsymbol{\phi}^s_{dn}\end{aligned}$$

where, $\hat{r}_{ud} = (\widehat{\mathbf{a}}_u \otimes \widehat{\mathbf{s}}_d)^T\boldsymbol{\theta} + \widehat{\mathbf{s}}_d^T(\boldsymbol{\gamma}_0 + \widehat{\boldsymbol{\gamma}}_u)$.

For the mean-field parameters $\{\boldsymbol{\phi}^a_u\}$,$\{\boldsymbol{\phi}^s_d\}$, minimizing the KL while enforcing normalization leads to the following solutions:

$$\begin{aligned}\phi^a_{unk} &= \frac{\beta_{k,w^a_{un}}\exp(\widehat{a}_{uk})}{\sum_j\beta_{j,w^a_{un}}\exp(\widehat{a}_{uj})} \\ \phi^s_{dmk} &= \frac{\beta_{k,w^s_{dm}}\exp(\widehat{s}_{dk})}{\sum_j\beta_{j,w^s_{dm}}\exp(\widehat{s}_{dj})}.\end{aligned}$$

We update the variational parameters of all users and subsequently of all documents (see Algorithm 1).

[5]While we could cycle through all variational parameters until convergence before beginning the M-step, we've found a single pass of updates per E-step to work well in practice.

**Algorithm 1** EM for the CSTM

**Input:** $\{\mathbf{w}_u^a\}, \{\mathbf{w}_d^s\}, \{r_{ud}\} \in \mathcal{O}$.

**while** Convergence criteria not met **do**
  # E-Step
  **for all** $d \in D$ **do**
    Update $\widehat{\mathbf{s}}_d, \boldsymbol{\phi}_d^s$
  **end for**
  **for all** $u = 1 \dots U$ **do**
    Update $\widehat{\mathbf{a}}_u, \widehat{\boldsymbol{\gamma}}_u, \boldsymbol{\phi}_u^a$
  **end for**
  # M-Step
  Update $\theta, \gamma_0, \boldsymbol{\beta}$
**end while**

## M-Step

The M-step aims to maximize the expectation of the complete likelihood under the variational posterior (taking into account the prior over the parameters $\boldsymbol{\gamma}_0, \boldsymbol{\theta}$):

$$\begin{aligned}
&\mathrm{E}_q\left[\mathcal{L}\right] + \log p(\boldsymbol{\gamma}_0) + \log p(\boldsymbol{\theta}) = \\
&\frac{1}{2\sigma_r^2} \sum_{(u,d)\in\mathcal{O}} \left(r_{ud} - ((\widehat{\mathbf{a}}_u \otimes \widehat{\mathbf{s}}_d)^T \boldsymbol{\theta} + (\boldsymbol{\gamma}_0 + \widehat{\boldsymbol{\gamma}}_u)^T \widehat{\mathbf{s}}_d)\right)^2 \\
&+ \sum_{u,n,k}^{U,N,K} \phi_{unk}^a \log \beta_{k,w_{un}^a} + \sum_{d,m,k}^{D,M,K} \phi_{dmk}^s \log \beta_{k,w_{dm}^s} \\
&- \frac{1}{2\lambda_{\gamma_0}^2} \boldsymbol{\gamma}_0^T \boldsymbol{\gamma}_0 - \frac{1}{2\lambda_\theta^2} \boldsymbol{\theta}^T \boldsymbol{\theta} + \text{constant} \,. \qquad (3)
\end{aligned}$$

Setting derivatives to zero (and satisfying the $\beta_{jw}$ parameters' normalization constraint), we obtain the following updates:

$$\begin{aligned}
\boldsymbol{\theta} =& -\frac{1}{\sigma_r}\Big(\sum_{(u,d)\in\mathcal{O}} (r_{ud} - \mathbf{s}_d^T(\boldsymbol{\gamma}_0 + \boldsymbol{\gamma}_u))(-s_d \otimes a_u)^T\Big) \\
&\left(\sum_{(u,d)\in\mathcal{O}} (s_d \otimes a_u)^2/\sigma_r + \lambda_\theta I/2\right)^{-1} \\
\boldsymbol{\gamma}_0 =& -\frac{1}{\sigma_r}\Big(\sum_{(u,d)\in\mathcal{O}} (r_{ud} - (s_d \otimes a_u)^T\theta - (\boldsymbol{\gamma}_u^T s_d))s_d\Big) \\
&\left(\sum_{(u,d)\in\mathcal{O}} \mathbf{s}_d \mathbf{s}_d^T/\sigma_r + \lambda_{\gamma_0} I/2\right)^{-1} \\
\beta_{jk} =& \frac{\sum_{u,n} \phi_{unj}^a 1_{\{w_{un}^a = k\}} + \sum_{d,m} \phi_{dmj}^s 1_{\{w_{dm}^s = k\}}}{\sum_{k',u,n} \phi_{unj}^a 1_{\{w_{un}^a = k'\}} + \sum_{d,m} \phi_{dmj}^s 1_{\{w_{dm}^s = k'\}}} \,.
\end{aligned}$$

At test time, prediction of missing preferences is made using $\hat{r}_{ud}$, which is readily available. That is we use the expectation of the variational variables to form estimates of $\hat{r}$

## 4. RELATED WORK

Previous work includes a few models that have combined item-only topic and regression models for user-item preference prediction. We are not aware of any earlier work that develops a text-based model of a user, nor one that combines user and item side information as in CSTM.

In [2] the authors model several sources of side information including item textual side information using LDA. The topic assignment proportions of documents ($\sum_m z_{dm}^s/M \; \forall d$) are used as item features and combined multiplicatively with user-demographic and behavioural features. The result is linearly combined with the other sources of side information to generate preferences.

In [20] the authors also combine LDA with a regression model for the task of recommending scientific articles. Here the item topic proportions are used as a prior mean on normally-distributed item (regression) latent variables. User latent variables are also normally distributed from a zero-mean prior. A specific user-item score is then generated as the inner product of item and user latent variables: $r_{ud} = \mathbf{a}_u^T(\mathbf{s}_d + \boldsymbol{\epsilon}_d)$, where $\boldsymbol{\epsilon}_d$ is drawn from a zero-mean normal. The preference prediction model is the same as the one used in probabilistic matrix factorization [15]. [20] also report that on their data a modified version of their model which is analogous to the model of [2] performed worse. [16] proposed a similar model without $\epsilon$ and used CTM [5]. For a similar application, [18] propose an approach based on link-prediction in a user-item graph based on user and item similarity as well as user (binary) preferences.

The fact that we model an additional type of information (user textual side information) makes it difficult to directly compare our model to the ones above. In addition, the parametrization we use to predict preferences is very different from previous models. We initially experimented with a parametrization similar to [20], albeit modified to also model user side information, and found it did not perform as well as CSTM (see Section 5 for an experimental comparison).

Finally, [1] propose a collaborative filtering model with side information. Although the form of the side information is not amenable to using topic models, the authors utilize a combination of linear models to obtain good performance in both cold and warm-start data regimes.

## 5. EXPERIMENTS

We first describe the three datasets used for experiments. We then introduce a set of methods for empirical comparisons, ranging from pure CF methods to pure side information methods. We report three separate sets of experiments. In the first we focus on cold-start users and examine the effect of including user libraries. In the second we study how the methods perform on users with varying amounts of observed scores. Finally, we design a synthetic paper recommendation experiment and simulate incoming users in order to test the value of both the user library and the user-provided item scores.

### 5.1 Datasets

We evaluate the models using these three datasets:

*Conf-1:* A dataset from the 2010 edition of the neural information procession systems (NIPS) conference. Users are conference reviewers while documents are the set of papers submitted to the conference. The dataset consists of 48 users and 1251 documents. Each user's library consists of his/her own previously published papers. Users have an average of 31 documents which are concatenated into one. After some basic preprocessing the length of the joint vocabulary was slightly over 18,000 words. In this dataset all users have expressed scores (integers between 0 and 3) for an average of 143 papers (std. 14).

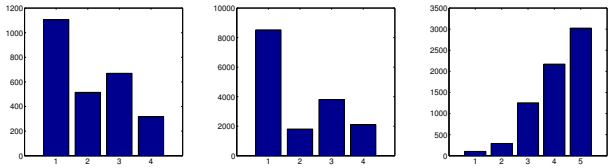

**Figure 2: Number of each rating values for the three datasets, from left to right: Conf-1, Conf-2, Books-1.**

| | User side-info. | Document side-info. | Shared Params |
|---|---|---|---|
| LM-I | ✓ | ✓ | |
| LM-II | ✓ | ✓ | |
| LR | | ✓ | |
| PMF | | | ✓ |
| CTR | | ✓ | ✓ |
| CSTM | ✓ | ✓ | ✓ |

**Table 1: A comparison of the modelling capabilities of each model. "Shared Params" stands for models that share information in-between users and/or items (in other words those which use some form of CF).**

*Conf-2:* A second dataset is from the international conference on machine learning (ICML 2012). This dataset consists of 433 users (reviewers) and 861 documents (submissions). Users have an average of 25 documents (std. 29) each and the length of the joint vocabulary is 16,201 words. In this dataset the average number of expressed scores per user is 48 (std. 25).

*Books-1:* The third dataset is from a large North American online book retailer.[6] It contains 316 users and 2601 documents (books). Users average 81 documents (std. 100). We removed very-infrequent and very-frequent words (those appearing in less than 1% or more than 95% of all documents). The resulting vocabulary contains 6,440 words. Users have a minimum of 15 expressed scores (mean 22, std. 6).

For each dataset the number of available preference values is shown in Figure 2. We note that the size of our datasets, and not the computational cost of learning in our model, limits our ability to scale up. In fact, learning CSTM on our largest dataset takes on the order of 2 hours on a modern machine using our *Matlab* implementation.

## 5.2 Competing models

We introduce several models which will serve as comparison to CSTM. Each model has particular characteristics (Table 1) which will help in understanding CSTM's performance.

Note that we use topic representations of documents for all competing models that use side information. Such representations were learned using a correlated topic model offline [5]. We re-use some of our previous notation to describe these models. Namely $A_u$ and $S_d$ are $K$-length vectors which designate a user's and a document's (topic) representation respectively.

[6]Kobo: `http://www.kobo.com`

*Constant:* Model predicts the average observed preferences for all missing preferences. Comparison to this baseline is useful to evaluate the value of learning.

*LM-I:* This model is meant to be a supervised version of the language model (LM[12]): $\hat{r}_{ud} := (A_u^T\theta_A)(S_d^T\theta_S)^T$ where, the parameters, $\theta_A, \theta_S$ are $K \times F$ matrices. $F$ is a hyper-parameter determined using a validation set (ranges from 5 to 30).

*LM-II:* Uses *isotonic regression* (see for example [4]) to calibrate the LM. The idea is to learn a regression model that satisfies the implicit ranking established by the LM:

$$\text{minimize}_{\hat{\mathbf{r}}_u, \forall u} \sum_{(ud)\in\mathcal{O}} \left(\hat{r}_{ud} - r_{ud}\right)^2$$
$$\text{s.t. } \hat{r}_{ud} \leq \hat{r}_{u(d+1)}, \ \ \forall d.$$

where the constraints enforce a, user-specific, document ordering specified by the output of the LM. Once learned $\{\hat{r}\}$ are used as the model's predictions. To obtain predictions for an unobserved document we have found that using the average score given to the two (observed) documents ranked directly above and below the new document works well. The performed regression is user-specific and thus cannot be used for users with no observed preferences. For such users we simply re-use the learned parameters of its closest user (based on users' topic representations). A more principled approach, for example a collaborative one, lies outside of the scope of this paper.

*LR:* This is a user-specific regression model (see Section 2.1 for details) where predictions are given by: $r_{ud} = \boldsymbol{\gamma}_u^T S_d$.

*PMF [15]:* PMF is a state-of-the-art collaborative filtering approach. PMF's generative model postulates that users and documents live in a low-dimensional latent space, represented respectively by $U_u$ and $V_d$. A user-item preference is generated by taking the dot product between the corresponding user and item representations: $r_{ud} = U_u^T V_d$. The size of the latent space is determined using a validation set (range from 1 to 30).

*Collaborative topic regression (CTR-CTM):* CTR, is matrix factorization with document-content model introduced in [20]. CTR was briefly reviewed in Section 4. We use a slightly different version than the one introduced by its authors. Namely, we have replaced LDA by CTM. Also, in our application since all user-item scores are given we use a single variance value over scores ($\sigma$).

For *LM-I*, *LR* and *PMF* learning is performed using MAP by assuming a Gaussian likelihood model and zero-mean Gaussian priors over the model's parameters. The priors' variance are determined using a validation set.

We investigated a few other models which we do not fully describe here. Of note: instead of modeling user libraries as side-information we used the documents of user libraries as explicitly (highly-)scored items. We experimented with various scoring schemes but none lead to consistent improvements over baselines. We also experimented with replacing directed topic models with an supervised extension of an undirected topic model [13]. Further we experimented with replacing both topic models by (unconstrained) probabilistic matrix factorization [17]. However, in both cases, initial experiments were not as promising.

## 5.3 Results

To run CSTM on the above datasets we first concatenated user documents (for example a researcher's previously pub-

lished papers) into a single document. To get user and item topic proportions we learned a CTM topic model [5] using the content of the items and then projected user documents into that (learned) topic space to obtain user topic proportions. We directly used these topics in LM-1,LM-II,LR. Further we used these topics as initialization in models which jointly learn topics and scores (CSTM,CTR). In all experiments we use 30 topics.

For training we create 5 folds from the available scores. Each fold is split into 80 percent observed and 20 percent test. We used the first fold to determine the hyper-parameters of the model. We report the average results over the five folds as well as the variance of this estimator.

We want to evaluate the performance of CSTM in settings where some users have no observed scores. The cold-start setting is of particular practical importance and one that should allow a good model to leverage the user's side-information. Accordingly, in our datasets we randomly selected one fourth of all users and removed all of their observed scores but kept their test scores. Further, for Conf-1 and Books-1, whose users have a more uniform number of ratings, we binned the remaining users (three quarters) uniformly into three *categories*. For Conf-1, users in each category had 15, 30 and 55 observed scores respectively. In each of the three categories 5 ratings per user were kept for validation. For Books-1 users in the first two categories had 8 and 10 scores while the scores of users in the last category were left untouched (5 scores per user were kept for validation). For Conf-2 since users are already naturally distributed into categories, we split the observed data into 25 percent validation and 75 percent train.

For the next two experiments, for each dataset, we train each model on all of the data but we divide our discussion into two parts. First we discuss cold-start users and after we examine the (other) user categories.

### 5.3.1 Cold-Start Data Regime

We first report the results for the completely cold-start data regime. For the cold-start users, it is difficult to calibrate the output of the model to the correct score range since only the users' side-information is available. The hope is that the models can use the side-information to get a better understanding of users preferences and discriminate between items of interest and other items. Accordingly we report results using *NDCG*. Normalized DCG (NDCG) is a well-established ranking measure, where a value of 1 indicates a perfect ranking and 0 a reverse-ordered perfect ranking [10]. NDCG@T considers exclusively the top $T$ items. Table 2 reports results for the three datasets using NDCG@5 (note that other values of NDCG gave similar results). We can only report results for the methods that have the ability to predict scores for cold-start users. PMF, LR and CTR do not use any user side-information and hence do not have that ability.

In this challenging setting CSTM significantly outperforms the other methods. Further we see that methods using side-information typically outperform the constant baseline. This demonstrates that the useful information about user preferences can be leveraged from the user libraries. Further, the good performance of CSTM in this setting shows that the model is able to leverage that information.

| | Conf-1 | Conf-2 | Books-1 |
|---|---|---|---|
| Constant | $0.4378 \pm 2e^{-3}$ | $0.6386 \pm 4e^{-5}$ | $0.6882 \pm 5e^{-4}$ |
| LM-I | $0.4684 \pm 2e^{-3}$ | $0.7903 \pm 4e^{-5}$ | $0.6873 \pm 1e^{-3}$ |
| LM-II | $0.4696 \pm 3e^{-4}$ | $0.7752 \pm 1e^{-4}$ | $0.6926 \pm 6e^{-4}$ |
| CSTM | $\mathbf{0.4846 \pm 1}e^{-3}$ | $\mathbf{0.8096 \pm 1}e^{-4}$ | $\mathbf{0.7360 \pm 2}e^{-4}$ |

**Table 2: Comparisons between CSTM and competitors for cold-start users using NDCG@5.**

### 5.3.2 Warm-start data regimes

The goal of CSTM is to perform well across different data regimes. In the previous section we examined models' performance on cold start users, we now focus on users with observed scores. For each dataset we report the performance of the various methods for each user category. For Conf-2 we separated users into roughly equal sized bins according to their number of observed scores. Results for our three datasets are provided in Figure 3. First we note that as the number of observed scores is increased the performance of the different methods also increases. CSTM outperforms all other methods on lower data-regimes. On users with more observed scores CSTM is competitive with both CTR-CTM and LR.

We notice that overall in this task, and even when many observed preferences are available, PMF is not competitive with most of the methods that have access to the side information. This highlights the value of content side-information on both user and item sides. This is further made clear by the relatively strong performance of both LM-I and LM-II.

Overall user libraries do not seem to help as much on the Books-1 dataset. There are several explanations for this. First, in Books-1 the distribution over scores is very skewed toward high scores. Thus a constant baseline does quite well. Further, bag-of-words representations are particularly well suited for academic papers where the presence (absence) of specific words are very good indications of the documents field and hence it's targeted audience. However, in (non-technical) books user preferences also rely on other aspects such as the document's prose which is harder to capture with a bag-of-words assumption.

### 5.3.3 Document Recommendations

We explore a different scenario which is meant to be closer to what would happen when a model is deployed in a complete recommendation system, for example to guide users to posters of interest in an academic conference. Specifically, we evaluate the performance of CTR and CSTM as new users arrive into the system and gradually provide information about themselves. We postulate that users first provide the system with their library. Then users gradually express their preferences for certain (user-chosen) items.

We trained CTR-CTM and CSTM on all but 50 randomly-chosen Conf-2 users with enough observed preferences (min. 15). We then simulated these users entering the system one by one. Since this experiment is about recommendations we report the results using NDCG.[7] Figure 4 presents the performance of CSTM and CTR-CTM as a function of the amount of data available in the system. When a user first

[7]In the absence of a recommendation objective and constraints it is reasonable to recommend the top-ranked items to each user

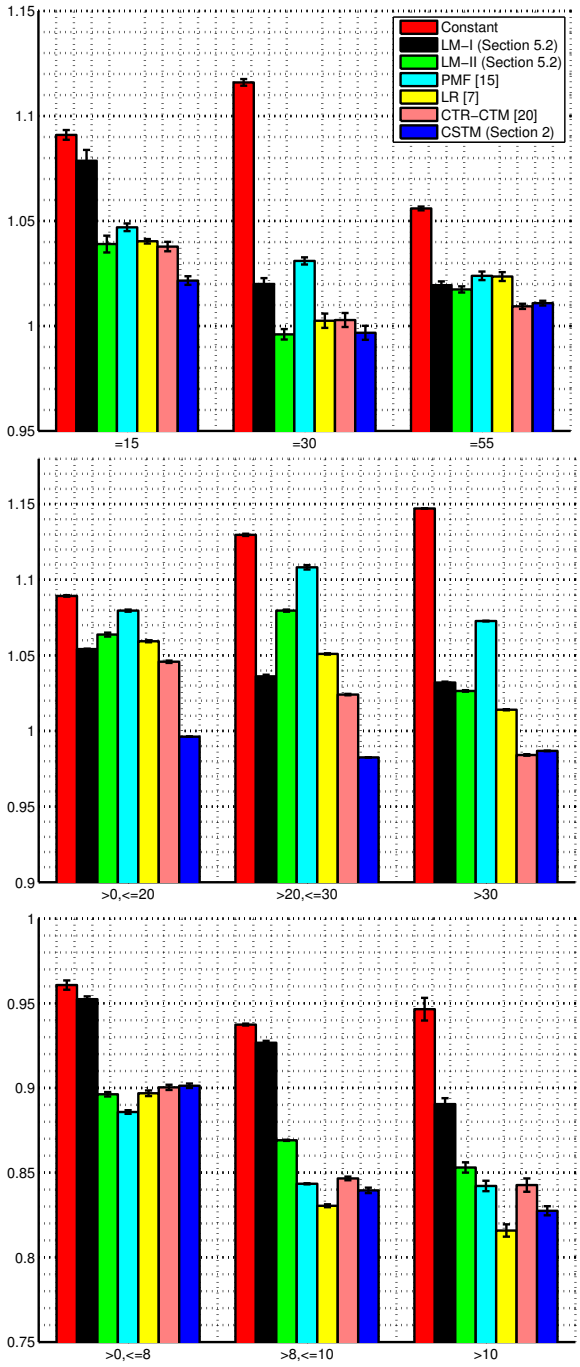


**Figure 3: Test RMSE of the different methods across Conf-1, Conf-2, Books-1. In each figure each group of bars reports results for different subsets of users. Each user is part of a single subset. The x-axis indicates the number of training observations per users of a given subset. The subsets correspond to users with the least observed preferences (left) to the most (right). Figures better seen in color (however the ordering in the legends corresponds to the ordering of the bars in each group).**

enters the system no data is available about her (indicated by "0" in the figure). The methods revert to using a constant predictor which predicts the mean of the previously observed scores across all users. Once a user provides a library (Lib.) we see that CSTM's performance increases very significantly. CTR cannot leverage that side information. Then once users provide scores, the performance of both methods increases and the performance of CTR eventually reaches the performance of CSTM.

Figure 4 demonstrates the advantage of having access to user side-information, namely, the system can quickly give good recommendations to new users. Further, in absolute terms the system performs relatively well without having access to any scores. It is also interesting to note that, in this experiment, as far as NDCG goes, the performance of CSTM only modestly improves as the number observed scores increases. This may be a consequence of our fairly primitive *online* learning procedure. As far as modelling goes this experiment is also a demonstration that our model of user libraries is effective at extracting features ($\mathbf{a}_u$ for all users) indicative of preferences and that the regression model then successfully combines the user and item side information.

As a practical experiment we deployed a system based on CSTM to a subset of the attendees of the most recent NIPS conference (NIPS-2013). NIPS is one of the most important machine learning conferences. We had previously gathered a dataset containing a few hundred of attendees' libraries and ratings. We also obtained text representations of the conference papers. Both sources of information were used to train CSTM. We used 20% of the observed ratings as a validation set used to determine the value of the hyper-parameters. Using the trained model we then generated predictions for each user using the same conditional inference procedure as above. We used each user's highest (predicted) ratings as their personalized recommendations. Furthermore, since NIPS 2013 had four daily poster sessions we allowed users to obtain independent paper recommendations for each day. A screen capture of the online user interface is provided in Figure 5.

We did not have a formal method of evaluating the quality or usefulness of the system beyond using the metrics we discussed in previous sections. Anecdotally, over 200 NIPS attendees accessed their recommendations and user feedback was almost unanimously positive. Furthermore we can explore the learned representations of the model as a way to assess its quality. Figure 6, shows the two-dimensional embeddings of user representations ($\mathbf{a}_u$) obtained using a popular (non-linear) dimensionality reduction technique for visualizing high-dimensional data [19]. We notice that, even in this low-dimensional representation, users cluster into different groups according to their areas of research. The model has discovered these groups using the similarities in user libraries and in their rating profiles. Similar results were obtained for paper representations.

## 6. CONCLUSION & FUTURE WORK

We have introduced a novel graphical model to leverage user libraries for preference prediction tasks. We showed experimentally that CSTM overall outperforms competing methods and can leverage the information of other users and of user libraries to perform particularly well in cold-start regimes. We also explored a paper recommendation

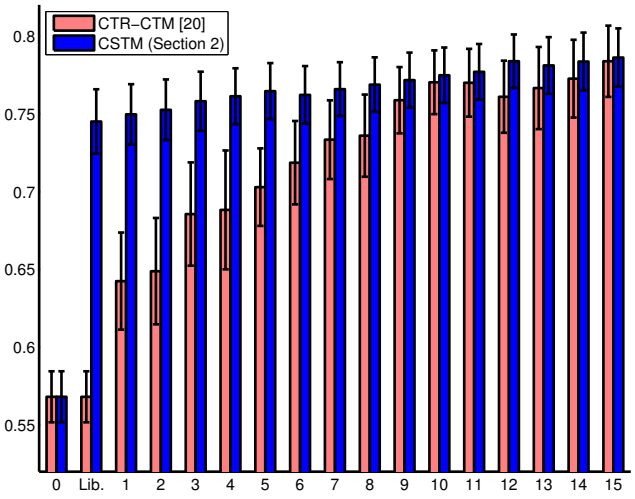


**Figure 4: Comparison of CSTM and CTR's NDCG@10 performance on new users as a function of the amount of data provided by users. Without any user data (0) methods revert to a constant predictor. Then CSTM takes advantage of user libraries (Lib.). Finally, scores are added one by one. Error bars indicate the variance across users.**

Menu: Top Papers | Thursday Papers | Friday Papers | Saturday Papers | Sunday Papers

1. Transfer Learning in a Transductive Setting (Sun81)
2. Deep Neural Networks for Object Detection (Sun78)
3. Visual Concept Learning: Combining Machine Vision and Bayesian Generalization on Concept Hierarchies (Sun76)
4. DeViSE: A Deep Visual-Semantic Embedding Model (Sun75)
5. Discriminative Transfer Learning with Tree-based Priors (Fri11)
6. Relevance Topic Model for Unstructured Social Group Activity Recognition (Thu51)
7. Deep Fisher Networks for Large-Scale Image Classification (Sun79)
8. Latent Maximum Margin Clustering (Sat01)
9. Latent Structured Active Learning (Thu25)
10. Structured Learning via Logistic Regression (Fri69)

**Figure 5: User interface of the NIPS-13 poster recommendations. The menu on the top of the page allows users of the system to obtain recommendations for one of the four daily poster sessions. Below the menu are the top-10 recommendations for a given user. Each recommendation links to the paper and contains the paper title and its unique conference identifier.**

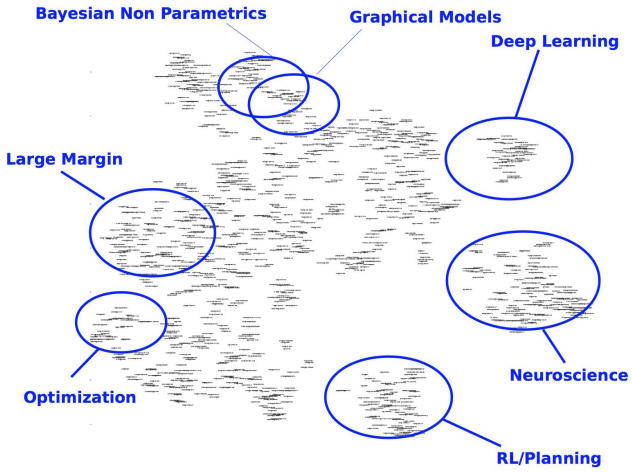


**Figure 6: We used t-SNE to obtain a two-dimensional representation of users. Each user is denoted using his or her email address. We note that users cluster according to easily identifiable subject areas some of which we have highlighted. A fully vectorized map is available at `http://www.cs.toronto.edu/~lcharlin/tmp/n13_tsne.pdf`.**

task and demonstrated the benefits of having access to user libraries.

Future work offers several possibilities. On one side we could refine the inference procedure used in training our model such as by using a fully variational approach or by leveraging the latest inference procedures of CTM [21]. Furthermore, it would also be straightforward to implement stochastic variational inference [9] for example by sampling users and updating relevant document and global parameters using a natural gradient. Stochastic inference is likely to be especially useful as we scale CSTM to very large datasets. On the other side we are examining extensions that enable the modelling of other types of side-information, such as book genres or academic-paper subject areas.

Another aspect of practical importance is that once we move to online recommendation, models must also be able to adapt to new data, including novel items and users, updates to user libraries, and new user-item scores. In the poster recommendations experiment we have seen that a simple *conditional inference* method works relatively well for novel users. However, one would also like to use the information from novel users to learn better representations of all users. In other words, we would need a mechanism which updates model parameters once a sufficient amount of new data is available. Furthermore, we could refine such a method *inter alia* to allow the system to adapt to the evolving preferences of users over time. For example, [1] propose a decaying mechanism to emphasize more recent scores over older ones. A similar mechanism could be use to weight the different documents in a user's library (for example based on date of publication for research papers or purchase date for books).

There is also the question of other potential applications for which CSTM could be useful. In addition to modelling text, topic models have also been shown to model images [8]. CSTM could then be used as an image recommendation tool

(for example to photographers). In that case, much like for the books of the Books-1 dataset, it remains to be seen whether topic models can capture features of images which are indicative of preferences.

**Acknowledgments:** We thank the NIPS'10 and ICML'12 program chairs for allowing us to use their data as well as Kobo for providing us with a useful dataset and support. We also acknowledge the support of CIFAR and NSERC.

## 7. REFERENCES

[1] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 19–28, New York, NY, USA, 2009. ACM.

[2] D. Agarwal and B.-C. Chen. flda: matrix factorization through latent dirichlet allocation. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 91–100, New York, NY, USA, 2010. ACM.

[3] R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *SIGKDD Explorations*, 9(2):75–79, 2007.

[4] M. J. Best and N. Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Math. Program.*, 47:425–439, 1990.

[5] D. M. Blei and J. D. Lafferty. A correlated topic model of science. *AAS*, 1(1):17–35, 2007.

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

[7] L. Charlin, R. Zemel, and C. Boutilier. A framework for optimizing paper matching. In *Proceedings of the Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pages 86–95, Corvallis, Oregon, 2011. AUAI Press.

[8] P. P. E. Bart, M. Welling. Unsupervised organization of image collections: Taxonomies and beyond. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 2011.

[9] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347, May 2013.

[10] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 41–48, New York, NY, USA, 2000. ACM.

[11] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, Nov. 1999.

[12] D. M. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In P. Berkhin, R. Caruana, and X. Wu, editors, *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 500–509, San Jose, California, 2007. ACM.

[13] R. Salakhutdinov and G. Hinton. Replicated softmax: an undirected topic model. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 1607–1614. 2009.

[14] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, volume 25, pages 880–887, Helsinki, Finland, 2008.

[15] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 1257–1264. MIT Press, Cambridge, MA, 2008.

[16] H. Shan and A. Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 1025–1030, Washington, DC, USA, 2010. IEEE Computer Society.

[17] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 650–658, New York, NY, USA, 2008. ACM.

[18] G. Tian and L. Jing. Recommending scientific articles using bi-relational graph-based iterative rwr. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, 2013.

[19] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, Nov. 2008.

[20] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 448–456, New York, NY, USA, 2011. ACM.

[21] C. Wang and D. M. Blei. Variational inference in nonconjugate models. *Journal of Machine Learning Research*, 14(1):1005–1031, Apr. 2013.

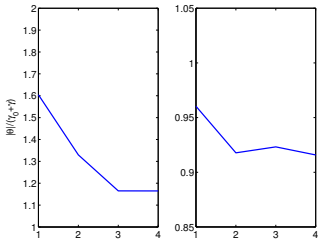

**Figure 7: Averaged norm of parameters under users with varying number of scores (left Conf-1, right Conf-2).**

| | Conf-2 |
|---|---|
| Constant | $0.8950 \pm 5e^{-4}$ |
| LM-I | $0.9301 \pm 2e^{-4}$ |
| LM-II | $0.9308 \pm 4e^{-4}$ |
| PMF | $0.9100 \pm 3e^{-4}$ |
| LR | $0.9288 \pm 2e^{-4}$ |
| CTR | $0.9375 \pm 3e^{-4}$ |
| CSTM | $\mathbf{0.9409 \pm 3}e^{-4}$ |

**Table 3: For the un-modified Conf-2 dataset, comparisons between CSTM and competitors for cold-start users using NDCG@5.**

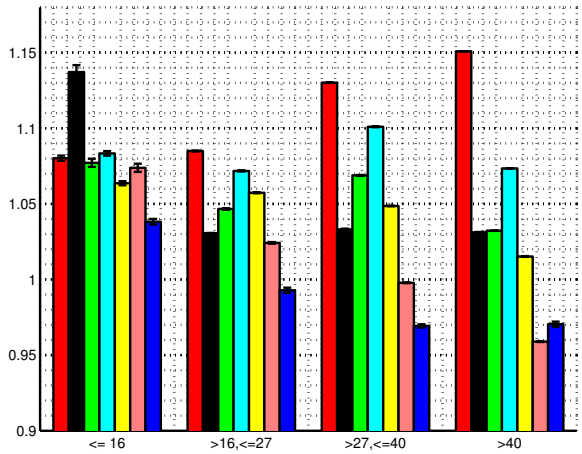


**Figure 8: RMSE results on the un-modified Conf-2 dataset.**

# APPENDIX

## A. ADDITIONAL RESULTS

### A.1 Model's Tradeoff

In Section 2 we motivated the specific parametrization of CSTM by its ability to trade off the influence of the user library side information versus that of the user-item scores. Here we show that learning in our model performs as expected. Figure 7 reports the relative norm of $\boldsymbol{\theta}$ versus $(\boldsymbol{\gamma}_0 + \boldsymbol{\gamma})$ as a function of the number of observed scores. As hypothesized as the number of scores increases the relative weight of the user library side information decreases.

### A.2 Variations of CSTM

We also experimented with variations of CSTM to better understand the roles played by the different aspects of the model and its training.

*CSTM fixed topics (CSTM-FT):* This model uses the exact preference regression model used by CSTM but it uses fixed topic user and document topic representations. That is it predicts preferences with: $r_{ud} = (\mathbf{a}_u \otimes \mathbf{s}_d)\theta^T + \mathbf{s}_d(\boldsymbol{\gamma}_0 + \boldsymbol{\gamma})^T$ where $A$ and $S$ are previously learned offline.

*CSTM no user side information (CSTM-NUSI):* To evaluate the gain of using user side information we experimented with a version of our model that does not model user side information (i.e., as if a user did not have any documents). Specifically, in this model $a_u \equiv \mathbf{0} \ \ \forall u$.

We provide some results comparing CSTM with its variations in table 4.

### A.3 Results on original Conf-2 dataset

In Figure 8 and Table 3 we provide comparisons of the different methods on the original version of Conf-2.

| | Conf-1 | Books-1 |
|---|---|---|
| CSTM-NUSI | $0.4941 \pm 4e^{-4}$ | $0.7997 \pm 8e^{-5}$ |
| CSTM-FT | $0.4984 \pm 2e^{-4}$ | $0.8026 \pm 8e^{-5}$ |
| CSTM | $\mathbf{0.5016 \pm 2}e^{-4}$ | $\mathbf{0.8037 \pm 2}e^{-5}$ |

| | Conf-2 | Conf-2(un-modified) |
|---|---|---|
| CSTM-NUSI | $0.7765 \pm 9e^{-5}$ | $0.8048 \pm 9e^{-4}$ |
| CSTM-FT | $0.8036 \pm 5e^{-5}$ | $0.8066 \pm 8e^{-5}$ |
| CSTM | $\mathbf{0.8217 \pm 2}e^{-5}$ | $\mathbf{0.8322 \pm 2}e^{-4}$ |

**Table 4: Comparisons between CSTM and two variations. Results report NDCG@5 over all users.**