ABSTRACT

Collaborative filtering analyzes user preferences for items (e.g., books, movies, restaurants, academic papers) by exploiting the similarity patterns across users. In implicit feedback settings, all the items, including the ones that a user did not consume, are taken into consideration. But this assumption does not accord with the common sense understanding that users have a limited scope and awareness of items. For example, a user might not have heard of a certain paper, or might live too far away from a restaurant to experience it. In the language of causal analysis [9], the assignment mechanism (i.e., the items that a user is exposed to) is a latent variable that may change for various user/item combinations. In this paper, we propose a new probabilistic approach that directly incorporates user exposure to items into collaborative filtering. The exposure is modeled as a latent variable and the model infers its value from data. In doing so, we recover one of the most successful state-of-the-art approaches as a special case of our model [8], and provide a plug-in method for conditioning exposure on various forms of exposure covariates (e.g., topics in text, venue locations). We show that our scalable inference algorithm outperforms existing benchmarks in four different domains both with and without exposure covariates.

Keywords

Recommender systems; collaborative filtering; matrix factorization

1. INTRODUCTION

Making good recommendations is an important problem on the web. In the recommendation problem, we observe how a set of users interacts with a set of items, and our goal is to show each user a set of previously unseen items that she will like. Broadly speaking, recommendation systems use historical data to infer users’ preferences, and then use the inferred preferences to suggest items. Good recommendation systems are essential as the web grows; users are overwhelmed with choice.

Traditionally there are two modes of this problem, recommendation from explicit data and recommendation from implicit data. With explicit data, users rate some items (positively, negatively, or along a spectrum) and we aim to predict their missing ratings. This is called explicit data because we only need the rated items to infer a user’s preferences. Positively rated items indicate types of items that she likes; negatively rated items indicate items that she does not like. But, for all of its qualities, explicit data is of limited use. It is often difficult to obtain.

The more prevalent mode is recommendation from implicit data. In implicit data, each user expresses a binary decision about items—for example this can be clicking, purchasing, viewing—and we aim to predict unclicked items that she would want to click on. Unlike ratings data, implicit data is easily accessible. While ratings data requires action on the part of the users, implicit data is often a natural byproduct of their behavior, e.g., browsing histories, click logs, and past purchases.

But recommendation from implicit data is also more difficult than its explicit counterpart. The reason is that the data is binary and thus, when inferring a user’s preferences, we must use unclicked items. Mirroring methods for explicit data, many methods treat unclicked items as those a user does not like. But this assumption is mistaken, and overestimates the effect of the unclicked items. Some of these items—many of them, in large-scale settings—are unclicked because the user didn’t see them, rather than because she chose not to click them. This is the crux of the problem of analyzing implicit data: we know users click on items they like, but we do not know why an item is unclicked.

Existing approaches account for this by downweighting the unclicked items. In Hu et al. [8] the data about unclicked items are given a lower “confidence”, expressed through the variance of a Gaussian random variable. In Rendle et al. [25], the unclicked items are artificially subsampled at a lower rate in order to reduce their influence on the estimation. These methods are effective, but they involve heuristic alterations to the data.

In this paper, we take a direct approach to solving this problem. We develop a probabilistic model for recommendation called Exposure MF (abbreviated as ExpoMF) that separately captures whether a user has been exposed to an item from whether a user has ultimately decided to click
on it. This leads to an algorithm that iterates between estimating the user preferences and estimating the exposure, i.e., why the unclicked items were unclicked. When estimating preferences, it naturally downweights the unclicked items that it expected the user will like, because it imagines that she was not exposed to them.

Concretely, imagine a music listener with a strong preference for alternative rock bands such as Radiohead. Imagine that, in a dataset, there are some Radiohead tracks that this user has not listened to. There are different reasons which may explain unlistened tracks (e.g., the user has a limited listening budget, a particular song is too recent or is unavailable from a particular online service). According to that user’s listening history these unlistened tracks would likely make for good recommendations. In this situation our model would assume that the user does not know about these tracks—she has not been exposed to them—and downweight their (negative) contribution when inferring that user’s preferences.

Further, by separating the two sides of the problem, our approach enables new innovations in implicit recommendation models. Specifically, we can build models of users’ exposure that are guided by additional information such as item content, if exposure to the items typically happens via search, or user/item location, if the users and items are geographically organized.

As an example imagine a recommender system for diners in New York City and diners in Las Vegas. New Yorkers are only exposed to restaurants in New York City. From our model’s perspective, unvisited restaurants in New York are therefore more informative in deriving a New Yorker’s preferences compared to unvisited restaurants in Las Vegas. Accordingly for New York users our model will upweight unvisited restaurants in New York while downweighting unvisited Las Vegas restaurants.

We studied our method with user listening history from a music intelligence company, clicks from a scientific e-print server, user bookmarks from an online reference manager, and user checkins at venues from a location-based social network. In all cases, ExpoMF matches or surpasses the state-of-the-art method of Hu et al. [8]. Furthermore, when available, we use extra information to inform our user exposure model. In those cases using the extra information outperforms the simple ExpoMF model. Further, when using document content information our model also outperforms a method specially developed for recommending documents using content and user click information [30]. Finally, we illustrate the alternative-rock-listener and the New-York-dinner examples using real data fit with our models in Figure 2 and Figure 3.

This paper is organized as follows. We first review collaborative filtering models in Section 2. We then introduce ExpoMF in Section 3 and location and content specific models in the subsequent subsection. We draw connections between ExpoMF and causal inference as well as other recommendation research paths in Section 4. Finally we present an empirical study in Section 5.

2. BACKGROUND

Matrix factorization for collaborative filtering. User-item preference data, whether implicit or not, can be encoded in a user by item matrix. In this paper we refer to this user by item matrix as the click matrix or the consumption matrix. Given the observed entries in this matrix the recommendation task is often framed as filling in the unobserved entries. Matrix factorization models, which infer (latent) user preferences and item attributes by factorizing the click matrix, are standard in recommender systems [11]. From a generative modeling perspective they can be understood as first drawing user and item latent factors correspondingly, respectively, to user preferences and item attributes. Then drawing observations from a specific distribution (e.g., a Poisson or a Gaussian) with its mean parametrized by the dot product between the user and the item factors. Formally, Gaussian matrix factorization is [18]:

\[
\begin{align*}
\theta_u &\sim N(\mathbf{0}, \lambda^u I_K) \\
\beta_i &\sim N(\mathbf{0}, \lambda^i I_K) \\
y_{ui} &\sim N(\theta^u_i \beta_i, \lambda_y^{-1}),
\end{align*}
\]

where \(\theta_u\) and \(\beta_i\) represent user \(u\)’s latent preferences and item \(i\)’s attributes respectively. We use the mean and covariance to parametrize the Gaussian distribution. \(\lambda_u, \lambda_i\), and \(\lambda_y\) are treated as hyperparameters. \(I_K\) stands for the identity matrix of dimension \(K\).

Collaborative filtering for implicit data. Weighted matrix factorization (WMF), the standard factorization model for implicit data (also known as one-class collaborative filtering [20]), selectively downweights evidence from the click matrix [8]. WMF uses a simple heuristic where all unobserved user-item interactions are equally downweighted vis-a-vis the observed interactions. Under WMF an observation is generated from:

\[
y_{ui} \sim N(\theta_i^u \beta_i, c_{ui}^{-1}),
\]

where the “confidence” \(c\) is set such that \(c_1 > c_0\). This dependency between a click and itself is unorthodox; because of it WMF is not a generative model. As we will describe in Section 3 we obtain a proper generative model by adding an exposure latent variable.

WMF treats the collaborative filtering problem with implicit data as a regression problem. Concretely, consumed user-item pairs are assigned a value of one and unobserved user-item pairs are assigned a value of zero. Bayesian personalized ranking (BPR) [25, 24] instead treats the problem as a one of ranking consumed user-item pairs above unobserved pairs. In a similar vein, the weighted approximate-ranking pairwise (WARP) loss proposed in Weston et al. [31] approximately optimizes Precision@k. To deal with the non-differentiable nature of the ranking loss, these methods typically design specific (stochastic optimization) methods for parameter estimation.

3. EXPOSURE MATRIX FACTORIZATION

We present exposure matrix factorization (ExpoMF). In Section 3.1, we describe the main model. In Section 3.2 we discuss several ways of incorporating external information into ExpoMF (i.e., topics from text, locations). We derive inference procedures for our model (and variants) in Section 3.3. Finally we discuss how to make predictions given our model in Section 3.4.
3.1 Model Description

For every combination of users \( u = 1, \ldots, U \) and items \( i = 1, \ldots, I \), consider two sets of variables. The first matrix \( A = \{a_{ui}\} \) indicates whether user \( u \) has been exposed to item \( i \). The second matrix \( Y = \{y_{ui}\} \) indicates whether or not user \( u \) clicked on item \( i \).

Whether a user is exposed to an item comes from a Bernoulli. Conditional on being exposed, user’s preference comes from a matrix factorization model. Similar to the standard methodology, we factorize this conditional distribution to a matrix factorization model. Similar to the standard method of a user’s preferences not user \( u \) then the log likelihood of no clicks \( \log(y_{ui}) \) is given in Figure 1a.

\[
\begin{align*}
\theta_u & \sim N(0, \lambda_\theta^{-1} I_K) \\
\beta_i & \sim N(0, \lambda_\beta^{-1} I_K) \\
a_{ui} & \sim \text{Bernoulli}(\mu_{ui}) \\
y_{ui} | a_{ui} = 1 & \sim N(\theta_u^\top \beta_i, \lambda_y^{-1}) \\
y_{ui} | a_{ui} = 0 & \sim \delta_0,
\end{align*}
\]

where \( \delta_0 \) denotes that \( p(y_{ui} = 0 | a_{ui} = 0) = 1 \), and we introduced a set of hyperparameters denoting the inverse variance \( (\lambda_\theta, \lambda_\beta, \lambda_y) \). \( \mu_{ui} \) is the prior probability of exposure, we discuss various ways of setting or learning it in subsequent sections. A graphical representation of the model in Equation (1) is given in Figure 1a.

We observe the complete click matrix \( Y \). These have a special structure. When \( y_{ui} > 0 \), we know that \( a_{ui} = 1 \). When \( y_{ui} = 0 \), then \( a_{ui} \) is latent. The user might have been exposed to item \( i \) and decided not to click (i.e., \( a_{ui} = 1, y_{ui} = 0 \)); or she may have never seen the item (i.e., \( a_{ui} = 0, y_{ui} = 0 \)). We note that since \( Y \) is usually sparse in practice, most \( a_{ui} \) will be latent.

The model described in Equation (1) leads to the following log joint probability\(^1\) of exposures and clicks for user \( u \) and item \( i \),

\[
\log p(a_{ui}, y_{ui} | \mu_{ui}, \theta_u, \beta_i, \lambda_y^{-1}) = \log \text{Bernoulli}(a_{ui} | \mu_{ui}) + a_{ui} \log N(y_{ui} | \theta_u^\top \beta_i, \lambda_y^{-1}) + (1 - a_{ui}) \log [y_{ui} = 0],
\]

where \( \mathbb{I}[b] \) is the indicator function that evaluates to 1 when \( b \) is true, and 0 otherwise.

What does the distribution in Equation (2) say about the model’s exposure beliefs when no clicks are observed\(^2\)? When the predicted preference is high (i.e., when \( \theta_u^\top \beta_i \) is high) then the log likelihood of no clicks \( \log N(0 | \theta_u^\top \beta_i, \lambda_y^{-1}) \) is low and likely non-positive. This feature penalizes the model for placing probability mass on \( a_{ui} = 1 \), forcing us to believe that user \( u \) is not exposed to item \( i \). (The converse argument also holds for low values of \( \theta_u^\top \beta_i \).) Interestingly, a low value of \( a_{ui} \) downweights the evidence for \( \theta_u \) and \( \beta_i \) (this is clear by considering extreme values: when \( a_{ui} = 0 \), the user and item factors do not affect the log joint in Eq. 2 at all; when \( a_{ui} = 1 \), we recover standard matrix factorization). Like weighted matrix factorization (WMF) [5], ExpoMF shares the same feature of selectively downweighting evidence from the click matrix.

In ExpoMF, fixing the entries of the exposure matrix to a single value (e.g., \( a_{ui} = 1, \forall u, i \)) recovers Gaussian probabilistic matrix factorization [18] (see Section 2). WMF is also a special case of our model which can be obtained by fixing ExpoMF’s exposure matrix using \( c_0 \) and \( c_1 \) as above.

The intuitions we developed for user exposure from the joint probability do not yet involve \( \mu_{ui} \), the prior belief on exposure. As we noted earlier, there are a rich set of choices available in the modeling of \( \mu_{ui} \). We discuss several of these next.

3.2 Hierarchical Modeling of Exposure

We now discuss methods for choosing and learning \( \mu_{ui} \). One could fix \( \mu_{ui} \) at some global value for all users and items, meaning that the user factors, item factors, and clicks would wholly determine exposure (conditioned on variance hyperparameters). One could also fix \( \mu_{ui} \) for specific values of \( u \) and \( i \). This can be done when there is specific extra information that informs us about exposure (denoted as exposure covariates), e.g. the location of a restaurant, the content of a paper. However, we found that empirical performance is highly sensitive to this choice, motivating the need to place models on the prior for \( \mu_{ui} \) with flexible parameters.

We introduce observed exposure covariates \( x_i \) and exposure model parameters \( \psi_u \) and condition \( \mu_{ui} | \psi_u, x_i \) according to some domain-specific structure. The extended graphical model with exposure covariates is shown in Figure 1b. Whatever this exposure model looks like, conditional independence between the priors for exposure and the more standard collaborative filtering parameters (given exposure) ensures that the updates for the model we introduced in Section 3.1 will be the same for many popular inference procedures (e.g., expectation-maximization, variational inference, Gibbs sampling), making the extension to exposure covariates a plug-in procedure. We discuss two possible choices of exposure model next.

**Per-item \( \mu_i \).** A direct way to encode exposure is via item popularity: if a song is popular, it is more likely that you have been exposed to it. Therefore, we choose an item-
Text topics or locations as exposure covariates. In the domain of recommending text documents, we consider the exposure covariates as the set of words for each document. In the domain of location-based recommendation, the exposure covariates are the locations of the venues being recommended. We treat both in a similar way.

Consider a $L$-dimensional ($L$ does not necessarily equal the latent space dimension $K$ in the matrix factorization model) representation $x_i$ of the content of document $i$ obtained through natural language processing (e.g., word embeddings [17], latent Dirichlet allocation [2]), or the position of venue $i$ obtained by first clustering all the venues in the data set then finding the expected assignment to $L$ clusters for each venue. In both cases, $x_i$ is all positive and normalizes to 1. Denoting $\sigma$ as the sigmoid function, we set

$$\mu_{ui} = \sigma(\psi_u^T x_i),$$

where we learn the coefficients $\psi_u$ for each user $u$. Furthermore, we can include intercepts with various levels and interactions [6].

How to interpret the coefficients $\psi_u$? The first interpretation is that of logistic regression, where the independent variables are $x_i$, the dependent binary variables are $a_{ui}$, and the coefficients to learn are $\psi_u$.

The second interpretation is from a recommender systems perspective: $\psi_u$ represents the topics (or geographical points of interest) that a user is usually exposed to, restricting the choice set to documents and venues that match $\psi_u$. For example, if the $l^{th}$ topic represents neural networks, and $x_{i,l}$ is high, then the user must be an avid consumer of neural network papers (i.e., $\psi_{ul}$ must be high) for the model to include an academic paper $i$ in the exposure set of $u$. In the location domain if the $l^{th}$ cluster represents Brooklyn, and $x_{i,l}$ is high, then the user must live in or visit Brooklyn often for the model to include venues near there in the exposure set of $u$.

3.3 Inference

We use expectation-maximization (EM) [3] to find the maximum a posteriori estimates of the unknown parameters of the model. The algorithm is summarized in Algorithm 1.

The EM inference procedure for the basic model, ExpoMF, can be found by writing out the full log likelihood of the model, then alternating between finding the expectations of missing data (exposure) in the E(xpectation)-step and finding maximum of the likelihood with respect to the parameters in the M(aximization)-step. This procedure is analytical for our model because it is conditionally conjugate, meaning that the posterior distribution of each random variable is in the same family as its prior in the model.

Furthermore, as we mentioned in Section 3.2, conditional independence between the priors for $\mu_{ui}$ and the rest of the model (given $\mu_{ui}$) means that the update for the latent exposure variables and user and item factors are not altered for any exposure model we use. We present these general updates first.

Algorithm 1 Inference for ExpoMF

1: **input:** click matrix $Y$, exposure covariates $x_{1:I}$ (Topics, location domain if the
2: **random initialization:** user factors $\theta_{1:U}$, item factors
3: **while** performance on validation set increases do
4: Compute expected exposure $A$ (Equation (4))
5: Update user factors $\theta_{1:U}$ (Equation (5))
6: Update item factors $\beta_{1:I}$ (Equation (6))
7: ExpoMF with per-item $\mu_i$.
8: ExpoMF with exposure model $\mu_{ui} = \sigma(\psi_u^T x_i)$.
9: end while

E-step. In the E-step, we compute expectation of the exposure latent variable $E[a_{ui}]$ for all user and item combinations $(u, i)$ for which there are no observed clicks (recall that the presence of clicks $y_{ui} > 0$ means that $a_{ui} = 1$ deterministically),

$$E[a_{ui} | \theta_u, \beta_i, \mu_{ui}, y_{ui} = 0] = \frac{\mu_{ui} \cdot N(0 | \theta_u^T \beta_i, \lambda_y^{-1})}{\mu_{ui} \cdot N(0 | \theta_u^T \beta_i, \lambda_y^{-1}) + (1 - \mu_{ui})},$$

where $N(0 | \theta_u^T \beta_i, \lambda_y^{-1})$ stands for the probability density function of $N(\theta_u^T \beta_i, \lambda_y^{-1})$ evaluated at 0.

M-step. For notational convenience, we define $p_{ui} = E[a_{ui} | \theta_u, \beta_i, \mu_{ui}, y_{ui} = 0]$ computed from the E-step. Without loss of generality, we define $p_{ui} = 1$ if $y_{ui} = 1$. The update for the latent collaborative filtering factors is:

$$\theta_u \leftarrow (\lambda_y \sum_i p_{ui} \beta_i^T + \lambda_B I_K)^{-1} \sum_i \lambda_y p_{ui} y_{ui} \beta_i$$

$$\beta_i \leftarrow (\lambda_y \sum_u p_{ui} \theta_u \beta_u^T + \lambda_B I_K)^{-1} \sum_u \lambda_y p_{ui} y_{ui} \theta_u,$$

Inference for the Exposure Prior $\mu_{ui}$

We now present inference for the hierarchical variants of Exposure MF. In particular we highlight the updates to $\mu_{ui}$ under the various models we presented in Section 3.2.

Update for per-item $\mu_i$. Maximizing the log likelihood with respect to $\mu_i$ is equivalent to finding the mode of the complete conditional Beta($\alpha_1 + \sum_u p_{ui}, \alpha_2 + U - \sum_u p_{ui}$), which is:

$$\mu_i \leftarrow \frac{\alpha_1 + \sum_u p_{ui}}{\alpha_1 + \alpha_2 + U - \sum_u p_{ui}}$$

Update for exposure covariates (topics, location).

Setting $\mu_{ui} = \sigma(\psi_u^T x_i)$, where $x_i$ is given by pre-processing (topic analysis or clustering), presents us with the challenge of maximizing the log likelihood with respect to exposure model parameters $\psi_u$. Since there is no analytical solution for the mode, we resort to following the gradients of the log likelihood with respect to $\psi_u$:

$$\psi_u^{new} \leftarrow \psi_u + \eta \nabla \psi_u L,$$

for some learning rate $\eta$, where

$$\nabla \psi_u L = \frac{1}{2} \sum_i (p_{ui} - \sigma(\psi_u^T x_i)) x_i.$$

This can be computationally challenging especially for large item-set sizes $I$. Therefore, we perform (mini-batch)
stochastic gradient descent: at each iteration $t$, we randomly sub-sample a small batch of items $B_t$ and take a noisy gradient steps:

$$
\psi_{u}^{\text{new}} \leftarrow \psi_{u} + \eta_t \hat{g}_t
$$

for some learning rate $\eta_t$, where

$$
\hat{g}_t = \frac{1}{|B_t|} \sum_{i \in B_t} (p_{ui} - \sigma(\psi_u^T x_i)) x_i.
$$

For each EM iteration, we found it sufficient to do a single update to the exposure model parameter $\psi_u$ (as opposed to updating until it reaches convergence). This partial M-step [10] is much faster in practice.

Complexity and Implementation Details
A naive implementation of the weighted matrix factorization (WMF) [8] has the same complexity as ExpoMF in terms of updating the user and item factors. However, the trick that is used to speed up computations in WMF cannot be applied to ExpoMF due to the non-uniformness of the exposure latent variable $a_{ui}$. On the other hand, the factor updates are still independent across users and items. These updates can therefore easily be parallelized.

In ExpoMF’s implementation, explicitly storing the exposure matrix $A$ is impractical for even medium-sized datasets. As an alternative, we perform the E-step on the fly: only the necessary part of the exposure matrix $A$ is constructed for the updates of the user/item factors and exposure priors $\mu_{ui}$. As shown in Section 5, with parallelization and the on-the-fly E-step, ExpoMF can be easily fit to medium-to-large datasets.

### 3.4 Prediction

In matrix factorization collaborative filtering the prediction of $y_{ui}$ is given by the dot product between the inferred user and item factors $\theta_u^\top \beta_i$. This corresponds to the predictive density of ExpoMF $p(y_{ui} | Y)$ using point mass approximations to the posterior given by the EM algorithm. However, ExpoMF can also make predictions by integrating out the uncertainty from the exposure latent variable $a_{ui}$:

$$
\text{E}_\theta[y_{ui} | \theta_u, \beta_i] = \text{E}_\theta[\text{E}_\theta[y_{ui} | a_{ui}, \theta_u, \beta_i, \mu_{ui}]]
= \sum_{a_{ui} \in \{0, 1\}} \text{P}(a_{ui}) \text{E}_\theta[y_{ui} | \theta_u, \beta_i, a_{ui}]
= \mu_{ui} \cdot \theta_u^\top \beta_i.
$$

We experimented with both predictions in our study and found that the simple dot product works better for ExpoMF with per-item $\mu_i$ while $\text{E} [y_{ui} | \theta_u, \beta_i]$ works better for ExpoMF with exposure covariates. We provide further insights about this difference in Section 5.6.

### 4. RELATED WORK

In this section we highlight connections between ExpoMF and other similar research directions.

**Causal inference.** Our work borrows ideas from the field of causal inference [22, 9]. Causal inference aims at understanding and explaining the effect of one variable on another.

One particular aim of causal inference is to answer counterfactual questions. For example, “would this new recommendation engine increase user click through rate?” While online studies may answer such a question, they are typically expensive even for large electronic commerce companies. Obtaining answers to such questions using observational data alone (e.g., log data) is therefore of important practical interest [3, 13, 28].

We establish a connection with the potential outcome framework of Rubin [26]. In this framework one differentiates the assignment mechanism, whether a user is exposed to an item, from the potential outcome, whether a user consumes an item. In potential outcome terminology our work can thus be understood as a form a latent assignment model. In particular, while consumption implies exposure, we do not know which items users have seen but not consumed. Further the questions of interest to us, personalized recommendation, depart from traditional work in causal inference which aims at quantifying the effect of a particular treatment (e.g., the efficacy of a new drug).

**Biased CF models.** Authors have recognized that typical observational data describing user rating items is biased toward items of interest. Although this observation is somewhat orthogonal to our investigation, models that emerged from this line of work share commonalities with our approach. Specifically, Marlin et al. [16], Ling et al. [14] separate the selection model (the exposure matrix) from the data model (the matrix factorization). However, their interpretation, rooted in the theory of missing data [15], leads to a much different interpretation of the selection model. They hypothesize that the value of a rating influences whether or not a user will report the rating (this implicitly captures the effect that users mostly consume items they like a priori). This approach is also specific to explicit feedback data. In contrast, we model how (the value of) the exposure matrix affects user rating or consumption.

**Modeling exposure with random graphs.** The user-item interaction can also be encoded as a bipartite graph. Paquet and Koenigstein [21] model exposure using a hidden consider graph. This graph plays a similar role as our exposure variable. One important difference is that during inference, instead of directly inferring the posterior as in ExpoMF (which is computationally more demanding), an approximation is developed whereby a random consider graph is stochastically sampled.

**Exposure in other contexts.** In zero-inflated Poisson regression, a latent binary random variable, similar to our exposure variable is introduced to “explain away” the structural zeros, such that the underlying Poisson model can better capture the count data [12]. This type of model is common in Economics where it is used to account for overly frequent zero-valued observations.

ExpoMF can also be considered as an instance of a spike-and-slab model [10] where the “spike” comes from the exposure variables and the matrix factorization component forms the flat “slab” part.

**Versatile CF models.** As we show in Section 3.2, ExpoMF’s exposure matrix can be used to model external in-
scientific articles data from arXiv taste profile subset (TPS) of the million song dataset [1]; 2) 5.1 Datasets exploring the resulting model fits. We highlight that: provide further insights into ExpoMF’s performance by ex-
of ExpoMF by fitting the model to several datasets. We
interaction between all features and learns specific parame-
meric) user, item or user-item features. FM considers the
machines (FM) of Rendle [23]. FM models all types of (nu-
dation scientific papers [30]). An exception is factorization
single type of data (e.g., document content when recommen-
dation model [30].)fmates outperforms a state-of-the-art document recom-
item
ates both outperform the simpler ExpoMF with per-
location covariates and ExpoMF with content covari-
 attributes according to
found that performance was improved by predicting missing
Further when using ExpoMF with exposure covariates we
calculate all the metrics based on the resulting ordered list.
Further when using ExpoMF with exposure covariates we
found that performance was improved by predicting missing preferences according to \( E[y_{ui}|\theta_u, \beta_i] \) (see Section 3.4 for
details).
5.3 Performance measures
To evaluate the recommendation performance, we report both Recall@k, a standard information retrieval measure, as well as two ranking-specific metrics: mean average precision (MAP@k) and NDCG@k.8

<table>
<thead>
<tr>
<th>TPS</th>
<th>Mendeley</th>
<th>Gowalla</th>
<th>ArXiv</th>
</tr>
</thead>
<tbody>
<tr>
<td># users</td>
<td>221,830</td>
<td>45,293</td>
<td>57,629</td>
</tr>
<tr>
<td># items</td>
<td>22,781</td>
<td>76,237</td>
<td>47,198</td>
</tr>
<tr>
<td># interactions</td>
<td>14.0M</td>
<td>2.4M</td>
<td>2.3M</td>
</tr>
<tr>
<td>% interactions</td>
<td>0.29%</td>
<td>0.07%</td>
<td>0.09%</td>
</tr>
</tbody>
</table>

Table 1: Attributes of datasets after pre-processing. Inter-
actions are non-zero entries (listening counts, clicks, and
checkins). % interactions refers to the density of the user-
item consumption matrix (\( Y \)).

formation describing user and item interactions. This is in
contrast to most CF models which are crafted to model a
single type of data (e.g., document content when recommen-
dation scientific papers [30]). An exception is factorization
interaction into training/test/validation sets with 70/20/10 propor-
tions. In all the experiments, the dimension of the latent space for collaborative filtering model \( K \) is 100. The
model is trained following the inference algorithm described in
Section 3.3. We monitor the convergence of the algorithm
using the truncated normalized discounted cumulative gain
(NDCG@100, see below for details) on the validation set.

5.2 Experimental setup
For each dataset we randomly split the observed user-item
interactions into training/test/validation sets with 70/20/10 propor-
tions. We do not use this metric because MPR penalizes ranks
linearly – if one algorithm ranks all the heldout test items at the top while the other half
at the bottom, both algorithms will get similar MPR, while
clearly the second algorithm is more desirable. On the other

• Mendeley: contains user-paper bookmarks as provided
by the Mendeley service, a “reference manager”. The
behavior data is filtered such that each user has at
least 10 papers in her library and the papers that are
bookmarked by at least 20 users are kept. In addition
this dataset contains the content of the papers which we
pre-process using standard techniques to yield a
10K words vocabulary. In Section 5.6 we make use of
paper content to inform ExpoMF’s exposure model.

• Gowalla: contains user-venue checkins from a location-
based social network. We pre-process the data such
that all users and venues have a minimum of 20 check-
ins. Furthermore, this dataset also contains locations
for the venues which we will use to guide location-
based recommendation (Section 5.6).

The final dimensions of these datasets are summarized in
Table 1.

5.1 Datasets
Throughout this study we use four medium to large-scale
user-item consumption datasets from various domains: 1) taste profile subset (TPS) of the million song dataset [1]; 2) scientific articles data from arXiv; 3) user bookmarks from Mendeley; and 4) check-in data from the Gowalla dataset.

• Taste Profile Subset (TPS): contains user-song play
counts collected by the music intelligence company Echo
Nest.\(^7\) We binarize the play counts and interpret them
as implicit preference. We further pre-process the dataset
by only keeping the users with at least 20 songs in their
listening history and songs that are listened to by at
least 50 users.

• ArXiv: contains user-paper clicks derived from log
data collected in 2012 by the arXiv pre-print server.
Multiple clicks by the same user on a single paper are
considered to be a single click. We pre-process the data
to ensure that all users and items have a minimum of
10 clicks.

To make predictions, for each user \( u \), we rank each item
using its predicted preference \( y_{ui} = \theta_u^T \beta_i \), \( i = 1, \ldots, I \). We
then exclude items from the training and validation sets and calculate all the metrics based on the resulting ordered list.

\[ \text{MPR} = \frac{\sum_{u,i} y_{ui}^\text{test} \cdot \text{perc}(u,i)}{\sum_{u,i} y_{ui}^\text{test}} \]

We do not use this metric because MPR penalizes ranks
linearly – if one algorithm ranks all the heldout test items in the middle of the list, while another algorithm ranks half
of the heldout test items at the top while the other half
at the bottom, both algorithms will get similar MPR, while
clearly the second algorithm is more desirable. On the other

\(^7\)http://arxiv.org
\(^8\)http://mendeley.com
\(^9\)http://the.echonest.com
We denote rank\((u, i)\) as the rank of item \(i\) in user \(u\)'s predicted list and \(y_{u,i}^{test}\) as the set of items in the heldout test set for user \(u\).

- **Recall@k**: For each user \(u\), Recall@\(k\) is computed as follows:
  \[
  \text{Recall}@k = \sum_{i \in y_{u}^{test}} \frac{1 \{\text{rank}(u, i) \leq k\}}{\min(k, |y_{u}^{test}|)}
  \]
  where \(\{ \cdot \}\) is the indicator function. In all our experiments we report both \(k = 20\) and \(k = 50\). We do not report Precision@\(k\) due to the noisy nature of the implicit feedback data: even if an item \(i \notin y_{u}^{test}\), it is possible that the user will consume it in the future. This makes Precision@\(k\) less interpretable since it is prone to fluctuations.

- **MAP@k**: Mean average precision calculates the mean of users’ average precision. The (truncated) average precision for user \(u\) is:
  \[
  \text{Average Precision}@k = \frac{\sum_{n=1}^{k} \text{Precision}@n}{\min(k, |y_{u}^{test}|)}
  \]

- **NDCG@k**: Emphasizes the importance of the top ranks by logarithmically discounting ranks. NDCG@\(k\) for each user is computed as follows:
  \[
  \text{DCG}@k = \frac{\sum_{i=1}^{k} 2^{\text{rel}_i} - 1}{\log_2(i+1)}; \quad \text{NDCG}@k = \frac{\text{DCG}@k}{\text{IDCG}@k}
  \]
  IDCG@\(k\) is a normalization factor that ensures NDCG lies between zero and one (perfect ranking). In the implicit feedback case the relevance is binary: \(\text{rel}_i = 1\) if \(i \in y_{u}^{test}\), and 0 otherwise. In our study we always report the averaged NDCG across users.

For the ranking-based measure in all the experiments we set \(k = 100\) which is a reasonable number of items to consider for a user. Results are consistent when using other values of \(k\).

### 5.4 Baselines

We compare ExpoMF to weighted matrix factorization (WMF), the standard state-of-the-art method for collaborative filtering with implicit data [8]. WMF is described in Section 2.

We also experimented with Bayesian personalized ranking (BPR) [25], a ranking model for implicit collaborative filtering. However preliminary results were not competitive with hand, the metrics we use here will clearly prefer the second algorithm.

other approaches. BPR is trained using stochastic optimization which can be sensitive to hyper-parameter values (especially hyper-parameters related to the optimization procedure). A more exhaustive search over hyper-parameters could yield more competitive results.

We describe specific baselines relevant to modeling exposure covariates in their dedicated subsections.

### 5.5 Studying Exposure MF

**Empirical evaluation.** Results comparing ExpoMF to WMF on our four datasets are given in Table 2. Each metric is averaged across all the users. We notice that ExpoMF performs comparably better than WMF on most datasets (the standard errors are on the order of \(10^{-4}\) though the difference in performance is small. In addition, higher values of NDCG@100 and MAP@100 (even when Recall@50 is lower) indicate that the top-ranked items by ExpoMF tend to be more relevant to users’ interests.

**Exploratory analysis.** We now explore posterior distributions of the exposure latent variables of two specific users from the TSP dataset. This exploration provides insights into how ExpoMF infers user exposure.

The top figure of Figure 2 shows the inferred exposure latent variable \(E[a_{ui}]\) corresponding to \(y_{ui} = 0\) for user A. \(E[a_{ui}]\) is plotted along with the empirical item popularity (measured by number of times a song was listened to in the training set). We also plot the interpolated per-item consideration prior \(\mu_i\) learned using Equation (7). There is a strong relationship between song popularity and consideration (this is true across users). User A’s training data revealed that she has only listened to songs from either Radiohead or Interpol (both are alternative rock bands). Therefore, for most songs, the model infers that the probability of user A considering them is higher than the inferred prior, i.e., it is more likely that user A did not want to listen to them (they are true zeros). However, as pointed out by the rectangular box, there are a few ‘outliers’ which mostly contain songs from Radiohead and Interpol that user A did not listen to (some of them are in fact held out in the test set). Effectively, a lower posterior \(E[a_{ui}]\) than the prior indicates that the model downweights these unlistened songs more. In contrast, WMF downweights all songs uniformly.

A second example is shown in the bottom figure of Figure 2. User B mostly listens to indie rock bands (e.g. Florence and the Machine, Arctic Monkeys, and The Kills). “Dog Days are Over” by Florence and the Machine is the second most popular song in this dataset, behind “Fireworks” by Katy Perry. These two songs correspond to the two rightmost dots on the figure. Given the user’s listening history, the model clearly differentiates these two similarly popular songs. The fact that user B did not listen to “Dog Days are
Incorporating Exposure Covariates

Figure 2 demonstrates that ExpoMF strongly associates user exposure to item popularity. This is partly due to the fact that the model’s prior is parametrized with a per-item term $\mu_i$.

Here we are interested in using exposure covariates to provide additional information about the (likely) exposure of users to items (see Figure 1b).

Recall that the role of these exposure covariates is to allow the matrix factorization component to focus on items that the user has been exposed to. In particular this can be done in the model by upweighting (increasing their probability of exposure) items that users were (likely) exposed to and downweighting items that were not. A motivating example is Las Vegas diners as discussed in Section 1.

In the coming subsections we compare content-aware and location-aware versions of ExpoMF which we refer to as Content ExpoMF and Location ExpoMF respectively. Studying each model in its respective domain we demonstrate that incorporating exposure covariates improve the quality of the recommendations compared to ExpoMF with per-item $\mu_i$.

Content Covariates

Scientists—whether through a search engine, a personal recommendation or other means—have a higher likelihood of being exposed to papers specific to their own discipline. In this section we study the problem of using the content of papers as a way to guide inference of the exposure component of ExpoMF.

In this use case, we model the user exposure based on the topics of articles. We use latent Dirichlet allocation (LDA) [2], a model of document collections, to model article content. Assuming there are $K$ topics $\Phi = \phi_1, \ldots, \phi_K$, each of which is a categorical distribution over a fixed set of vocabulary, LDA treats each document as a mixture of these topics where the topic proportion $x_i$ is inferred from the data. One can understand LDA as representing documents in a low-dimensional “topic” space with the topic proportion $x_i$ being their coordinates.

We use the topic proportion $x_i$ learned from the Mendeley dataset as exposure covariates. Following the notation of Section 3.2, our hierarchical ExpoMF is:

$$\mu_{ui} = \sigma(\psi_i \top x_i + \gamma_u)$$

where we include a per-user bias term $\gamma_u$. Under this model, a molecular biology paper and a computer science paper that a computer scientist has not read will likely be treated differently: the model will consider the computer scientist has been exposed to the computer science paper, thus higher $\mathbb{E}[\mu_{ui}]$, yet not to the molecular biology paper (hence lower $\mathbb{E}[\mu_{ui}]$). The matrix factorization component of the model will focus on modeling computer science papers since that are more likely to be have been exposed.

Our model, Content ExpoMF, is trained following the algorithm in Algorithm 1. For updating exposure-related model parameters $\psi_i$ and $\gamma_u$, we take mini-batch gradient steps with a batch-size of 10 users and a constant step size of 0.5 for 10 epochs.
Figure 3 highlights the behavior of Content ExpoMF compared to that of regular ExpoMF. Two users are selected: User A (left column) is interested in statistical machine learning and Bayesian statistics. User B (right column) is interested in computer systems. Neither of them have read “Latent Dirichlet Allocation” (LDA) a seminal paper that falls within user A’s interests. On the top row we show the posterior of the exposure latent variables $E[a_{ui}]$ for two users (user A and user B) inferred from ExpoMF with per-item $\mu_i$. LDA is shown using a white dot. Overall both users’ estimated exposures are dominated by the empirical item popularity.

In contrast, on the bottom row we plot the results of Content ExpoMF. Allowing the model to use the documents’ content to infer user exposure offers greater flexibility compared to the simple ExpoMF model. This extra flexibility may also explain why there is an advantage in using inferred exposure to predict missing observations (see Section 3.4). Namely when exposure covariates are available the model can better capture the underlying user exposures to items. In contrast using the inferred exposure to predict with the simple ExpoMF model performs worse.

**Location Covariates**

When studying the Gowalla dataset we can use venue location as exposure covariates.

Recall from Section 3.2 that location exposure covariates are created by first clustering all venues (using K-means) and then finding the representation of each venue in this clustering space. Similarly as in Content ExpoMF (Section 5.6), Location ExpoMF departs from ExpoMF:

$$\mu_{ui} = \sigma(\psi_i^T x_i + \gamma_u)$$

---

9Note that to train CTR we first learned a document topic model, fixed it and then learned the user preference model. It was suggested by its authors that this learning procedure provided computational advantages while not hindering performance significantly [30].
where $x_{ik}$ is the venue $i$’s expected assignment to cluster $k$ and $\gamma_u$ is a per-user bias term.\footnote{We named Content ExpoMF and Location ExpoMF differently to make it clear to the reader that they condition on content and location features respectively. Both models are in fact mathematically equivalent.}

### Study.
We train Location ExpoMF following the same procedure as Content ExpoMF. We report the empirical comparison between WMF, ExpoMF, and Location ExpoMF in Table 4. We note that Location ExpoMF outperforms both WMF and the simpler version of ExpoMF.

For comparison purposes we also developed a simple baseline FilterMF which makes use of the location covariates. FilterMF filters out venues recommended by WMF that are inaccessible (too far) to the user. Since user location is not directly available in the dataset, we estimate it using the geometric median of all the venues the user has checked into. The median is preferable to the mean because it is better at handling outliers and is more likely to choose a typical visit location. However, the results of this simple FilterMF baseline are worse than the results of the regular WMF. We attribute this performance to the fact that having a single focus of location is too strong an assumption to capture visit behavior of users well. In addition, since we randomly split the data, it is possible that a user’s checkins at city A and city B are split between the training and test set. We leave the exploration of better location-aware baselines to future work.

### 6. CONCLUSION

In this paper, we presented a novel collaborative filtering mechanism that takes into account user exposure to items. In doing so, we theoretically justify existing approaches that downweight unclicked items for recommendation, and provide an extendable framework for specifying more elaborate models of exposure based on logistic regression. In empirical studies we found that the additional flexibility of our model helps it outperform existing approaches to matrix factorization on four datasets from various domains. We note that the same approach can also be used to analyze explicit feedback.

There are several promising avenues for future work. Consider a reader who keeps himself up to date with the “what’s new” pages of a website, or a tourist visiting a new city looking for a restaurant recommendation. The exposure processes are more dynamic in these scenarios and may be different during training and test time. We therefore seek new ways to capture exposure that include ever more realistic assumptions about how users interact with items.

Finally, we would like to evaluate our proposed model in a more realistic setting, e.g., in an online environment with user interactions. It would be instructive to evaluate the performance of ExpoMF in environments where it may be possible to observe items which users have been exposed to.

### Acknowledgments

We thank Dustin Tran, Rajesh Ranganath, and reviewers for helpful feedback. This work is supported by IIS-1247664, ONR N00014-11-1-0651, DARPA FA8750-14-2-0009, Facebook, Adobe, Amazon, and the John Templeton Foundation.

### References


### Table 4: Comparison between Location ExpoMF and ExpoMF with per-item $\mu_i$ on Gowalla. Using location exposure covariates outperforms the simpler ExpoMF and WMF according to all metrics.

<table>
<thead>
<tr>
<th>Metric</th>
<th>WMF</th>
<th>ExpoMF</th>
<th>Location ExpoMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall@20</td>
<td>0.122</td>
<td>0.118</td>
<td>0.129</td>
</tr>
<tr>
<td>Recall@50</td>
<td>0.192</td>
<td>0.186</td>
<td>0.199</td>
</tr>
<tr>
<td>NDCG@100</td>
<td>0.118</td>
<td>0.116</td>
<td>0.125</td>
</tr>
<tr>
<td>MAP@100</td>
<td>0.044</td>
<td>0.043</td>
<td>0.048</td>
</tr>
</tbody>
</table>

We thank Dustin Tran, Rajesh Ranganath, and reviewers for helpful feedback. This work is supported by IIS-1247664, ONR N00014-11-1-0651, DARPA FA8750-14-2-0009, Facebook, Adobe, Amazon, and the John Templeton Foundation.


