Collaborative Prediction and Ranking with Non-Random Missing Data

Benjamin M. Marlin
Department of Computer Science
University of British Columbia
2366 Main Mall
Vancouver, Canada
bmarlin@cs.ubc.ca

Richard S. Zemel
Department of Computer Science
University of Toronto
6 King's College Road
Toronto, Canada
zemel@cs.toronto.edu

ABSTRACT

A fundamental aspect of rating-based recommender systems is the observation process, the process by which users choose the items they rate. Nearly all research on collaborative filtering and recommender systems is founded on the assumption that missing ratings are missing at random. The statistical theory of missing data shows that incorrect assumptions about missing data can lead to biased parameter estimation and prediction. In a recent study, we demonstrated strong evidence for violations of the missing at random condition in a real recommender system. In this paper we present the first study of the effect of non-random missing data on collaborative ranking, and extend our previous results regarding the impact of non-random missing data on collaborative prediction.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering; G.3 [**Probability and Statistics**]: Multivariate statistics; I.2.6 [**Learning**]: Parameter learning

General Terms

Algorithms, Performance

Keywords

recommender systems, collaborative filtering, ranking, probabilistic models, non-random missing data

1. INTRODUCTION

Collaborative filtering and recommender systems are principally concerned with two related problems: rating prediction and ranking. The goal of the rating prediction task is to accurately predict the rating a user would assign to an individual item. The goal of the ranking task is to provide the user with a personalized list of top ranked items. Collaborative filtering methods attempt to solve both problems by leveraging rating data collected from a large community of users.

Research on recommender systems has focused almost exclusively on properties of rating prediction and ranking methods including prediction accuracy [1], ranking accuracy [15], novelty and diversity [17, 16], and explainability [5]. In this paper, we focus on properties of the underlying rating observation process. We define the rating observation process as the process through which users select the items they choose to rate. In particular, we assess the impact of *non-random observation processes* on rating prediction and ranking.

Most prior research on collaborative filtering and recommender systems is founded on the assumption that missing ratings are missing at random. However, the statistical theory of missing data developed by Little and Rubin [6] shows that incorrect assumptions about missing data can lead to biased parameter estimation and prediction in a wide range of models and methods including clustering [1], matrix factorization [2] and other probabilistic models [11].

The key concept in Little and Rubin's theory is the *missing at random condition*. This condition is quite intuitive in the collaborative filtering setting. It essentially states that the probability that a rating is missing does not depend on the value of that rating, or the value of any other missing rating. The condition is easily violated in recommender systems if, for example, users are more likely to supply ratings for items that they do like, and less likely to supply ratings for items that they do not like. We recently reported substantial evidence for violations of the missing at random condition in recommender systems based on an online study conducted at Yahoo! Research involving over $35,000$ participants [8].

The main contribution of this paper is the first empirical analysis of ranking in the presence of non-random missing data. We also extend our previous investigation into the effect of non-random missing data on rating prediction by testing two additional baseline methods: nearest neighbour regression and matrix factorization. Our results show that two methods that incorporate a non-random missing data model, MM/CPT-v and MM/Logit-vd, outperform the baseline methods when evaluating rating prediction and ranking on items *selected at random*. We believe this is a more accurate measure of performance than testing on items selected by the user since it better reflects an important goal of

RecSys'09, October 23–25, 2009, New York, New York, USA.

recommender systems: to make predictions and recommend items *that the user has not yet seen or rated*.

2. MISSING DATA THEORY

A collaborative filtering data set can be represented as a rectangular matrix $\mathbf{x}$ where each row in the matrix represents a user, and each column in the matrix represents an item. x_{nd} denotes the rating of user n for item d. Let N be the number of users in the data set, D be the number of items, and V be the number of rating values. To reason about the observation process, we require a representation for missing and observed rating values. We introduce a companion matrix of response indicators $\mathbf{r}$ where $r_{nd} = 1$ if x_{nd} is observed, and $r_{nd} = 0$ if x_{nd} is not observed. Hierarchical collaborative filtering models such as clustering often contain latent values that are never observed. We denote latent values associated with data case n by $\mathbf{z}_n$. We denote the corresponding random variables with capital letters.

Following Little and Rubin, we introduce a parametric joint probability distribution on the data $\mathbf{x}$, response indicators $\mathbf{r}$, and latent values $\mathbf{z}$ [6]. We adopt the factorization of the joint distribution of the data random variables $\mathbf{X}$, response indicator random variables $\mathbf{R}$, and latent variables $\mathbf{Z}$ shown in Equation 1. μ and θ are the parameters of the distribution.

$$P(\mathbf{R}, \mathbf{X}, \mathbf{Z}|\mu, \theta) = P(\mathbf{R}|\mathbf{X}, \mathbf{Z}, \mu)P(\mathbf{X}, \mathbf{Z}|\theta) \tag{1}$$

Little and Rubin refer to $P(\mathbf{R}|\mathbf{X}, \mathbf{Z}, \mu)$ as the missing data model and $P(\mathbf{X}, \mathbf{Z}|\theta)$ as the data model. $P(\mathbf{R}|\mathbf{X}, \mathbf{Z}, \mu)$ is what we have been referring to as the observation process. The intuition behind this factorization, under the additional assumption that data cases are independently and identically distributed, is that all of a user's ratings are first generated according to the data model $P(\mathbf{X}_n, \mathbf{Z}_n|\theta)$, and the missing data model $P(\mathbf{R}_n|\mathbf{X}_n, \mathbf{Z}_n, \mu)$ is then used to decide which ratings will be observed and which will be missing.

2.1 Types of Missing Data

Little and Rubin classify missing data into several types including missing completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR) [6, p. 14]. The MCAR condition is defined in Equation 2, and the MAR condition is defined in Equation 3. Under MCAR the response probability for an item or set of items cannot depend on the data values in any way. Under the MAR condition, the data vector is divided into a missing part $\mathbf{x}^{mis}$ and an observed part $\mathbf{x}^{obs}$ according to the value of $\mathbf{r}$ in question: $\mathbf{x} = [\mathbf{x}^{mis}, \mathbf{x}^{obs}]$. The intuition is that the probability of observing a particular response pattern can only depend on the elements of the data vector that are observed under that pattern. Both MCAR and MAR require the additional technical condition that the parameters μ and θ be distinct, and that they have independent prior distributions.

$$P_{mcar}(\mathbf{R}|\mathbf{X}, \mathbf{Z}, \mu) = P(\mathbf{R}|\mu) \tag{2}$$

$$P_{mar}(\mathbf{R}|\mathbf{X}, \mathbf{Z}, \mu) = P(\mathbf{R}|\mathbf{X}^{obs}, \mu) \tag{3}$$

Missing data is NMAR when the MAR condition fails to hold. The simplest reason for MAR to fail is that the probability of not observing a particular element of the data vector depends on the value of that element. In the collaborative filtering case this corresponds to the idea that the probability of observing the rating for a particular item depends on the user's rating for that item, which is quite natural.

2.2 Impact Of Missing Data

When missing data is missing at random, maximum likelihood inference based only on the observed data $\mathbf{x}^{obs}$ will be unbiased. We demonstrate this result in Equation 7. The key property of the MAR condition is that the response probabilities are independent of the missing data, allowing the complete data likelihood to be marginalized independently of the missing data model. However, when missing data is not missing at random, this important property fails to hold, and it is not possible to simplify the likelihood beyond Equation 4 [6, p. 219]. Ignoring the missing data mechanism will clearly lead to biased parameter estimates and biased predictions since an incorrect likelihood function is being used. For non-identifiable models such as mixtures, we will use the terms "biased" and "unbiased" in a more general sense to indicate whether the parameters are estimated under the correct likelihood function.

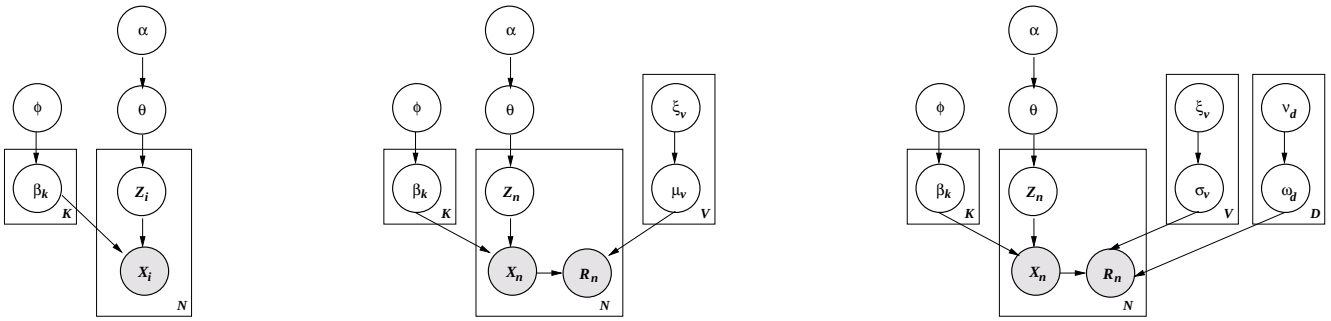
$$\begin{aligned}
&\mathcal{L}_{mar}(\theta|\mathbf{x}^{obs}, \mathbf{r}) \\
&= \int_{\mathbf{x}^{mis}} \int_{\mathbf{z}} P(\mathbf{X}, \mathbf{Z}|\theta) P(\mathbf{R}|\mathbf{X}, \mathbf{Z}, \mu) d\mathbf{Z} d\mathbf{X}^{mis} && (4)\\
&= P(\mathbf{R}|\mathbf{X}^{obs}, \mu) \int_{\mathbf{x}^{mis}} \int_{\mathbf{z}} P(\mathbf{X}, \mathbf{Z}|\theta) d\mathbf{Z} d\mathbf{X}^{mis} && (5)\\
&= P(\mathbf{R}|\mathbf{X}^{obs}, \mu) P(\mathbf{X}^{obs}|\theta) && (6)\\
&\propto P(\mathbf{X}^{obs}|\theta) && (7)
\end{aligned}$$

While this analysis of the impact of non-random missing data is based on maximum likelihood estimation, it immediately extends to the case of Bayesian inference. It also extends to learning in models like regularized matrix factorization that can be cast as probabilistic models [10]. In the case of non-parametric methods such as nearest neighbour regression, it may be possible to correctly identify relevant neighbours for a user or item in the presence of non-random missing data using common similarity measures like Pearson correlation [4]. However, it is clear that if missing data is not missing at random, the resulting rating predictions will be biased. Consider, for example, a case where low-valued ratings are more likely to be missing, this would create an over-abundance of high rating values in the observed data and the resulting rating predictions would be biased upward.

Unfortunately, the theory does not easily extend to the analysis of methods for learning to rank [14], or to the analysis of the ranking task itself. What we do know is that learning rating prediction models will be subject to bias in the presence of non-random missing data. Using these models for ranking in a standard score-and-sort framework is likely to pass some of this bias along to the inferred rankings, resulting in a degradation of ranking performance. The main question we are interested in is can we obtain better rating prediction and ranking performance using simple models of the non-random missing data process?

3. MODELS AND ALGORITHMS

The framework we consider for learning and prediction with non-random missing data follows the basic outline suggested by Little and Rubin [6]: We combine a probabilistic model for complete data, in this case a multinomial mixture



(a) Mixture Model (b) Mixture Model/CPT-v (c) Mixture Model/Logit-vd

Figure 1: Graphical models illustrating the basic multinomial mixture model, the multinomial mixture/CPT-v model, and the multinomial mixture/Logit-vd model.

clustering model, with a probabilistic model of the missing data process. The missing data models we consider capture some properties of a non-random missing data process, but are necessarily simplistic since our aim is to simultaneously estimate the parameters of both the complete data model and the missing data model. We begin by describing three models based on multinomial mixture clustering that incorporate different missing data assumptions. We also briefly describe the baseline matrix factorization and nearest neighbour methods.

3.1 Multinomial Mixture Model/MAR

The finite multinomial mixture model pictured in Figure 1(a) is a basic clustering model for discrete data. The random variables Z_n are cluster or mixture component indicator variables. They indicate which mixture component is associated with each data case and take values from the discrete set $\{1, ..., K\}$. The random variables Z_n are not observed and are referred to as latent variables. The mixing proportions θ_k give the prior probability of observing a data case from each of the K clusters. The parameters of the mixture component distributions are denoted by β_k. The component distributions are a product of independent multinomials where $\beta_{vdk} = P(x_{nd} = v|Z_n = k)$. We denote the prior distribution on the mixture component distributions β_k with hyper-parameters ϕ by $P(\beta_k|\phi)$. We denote the prior distribution on the mixing proportions θ with hyper-parameters α by $P(\theta|\alpha)$. We use independent Dirichlet distributions for both $P(\beta_k|\phi)$ and $P(\theta|\alpha)$. We give the probabilistic model for the multinomial mixture model in Equations 8 to 11. The square bracket notation $[s]$ represents an indicator function that takes the value 1 if the statement s is true, and 0 if the statement s is false.

$$P(\theta|\alpha) = \mathcal{D}(\theta|\alpha) \tag{8}$$

$$P(\beta_{dk}|\phi_{dk}) = \mathcal{D}(\beta_{dk}|\phi_{dk}) \tag{9}$$

$$P(Z_n = k|\theta) = \theta_k \tag{10}$$

$$P(\mathbf{x}_n|Z_n = k, \beta) = \prod_{d=1}^{D}\prod_{v=1}^{V} \beta_{vdk}^{[x_{nd}=v]} \tag{11}$$

Clustering models have a very natural interpretation in the collaborative filtering domain: the latent variable z_n indicates the group or cluster that user n belongs to, and the parameters β_k specify the preferences of a prototypical user that belongs to group k.

The default when dealing with missing data in a mixture model is to invoke the missing at random assumption. Under the missing at random assumption, the missing data model can be ignored, and inference, learning, and prediction can be based on the observed data only. In the multinomial mixture model, missing data can be analytically summed out of the observed data posterior.

3.2 Multinomial Mixture Model/CPT-v

The multinomial mixture model is a natural baseline model for collaborative filtering. It does not include an explicit missing data model, and hence relies on the MAR assumption to deal with missing data. We now review an extension of the multinomial mixture model that includes an explicit non-random missing data mechanism that we call *CPT-v* [8]. CPT-v is a simple missing data model where the probability that a rating is observed depends only on that underlying rating value. This model can capture the idea that a user's preferences for a particular item can influence whether the user rates that item, but the effect is the same for all items.

The CPT-v missing data model is parameterized using a conditional probability table consisting of V Bernoulli parameters μ_v (hence the name CPT-v). The parameters μ_v give the probability that an item will be rated if its true rating value is v: $P(r_{nd} = 1|x_{nd} = v) = \mu_v$. A Bayesian network representation of the combined finite multinomial mixture model/CPT-v model is given in Figure 1(b). The response indicators are assumed to be independently sampled for each item, leading to the missing model given in Equation 12. The prior distribution on the μ_v parameters is a Beta distribution as seen in Equation 13.

$$P(\mathbf{r}_n|\mathbf{x}_n, \mu) = \prod_{d=1}^{D}\prod_{v=1}^{V} (\mu_v^{[r_{nd}=1]}(1-\mu_v)^{[r_{nd}=0]})^{[x_{nd}=v]} \tag{12}$$

$$P(\mu|\xi) = \prod_{v=1}^{V} \mathcal{B}(\mu_v|\xi_{0v}, \xi_{1v}) \tag{13}$$

The specification of the multinomial mixture complete data model is given by Equations 8-11.

3.3 Multinomial Mixture Model/Logit-vd

The CPT-v missing data model is restrictive in that it asserts the same conditional missing data rates for all items. We now present a more flexible missing data model that we call *Logit-vd*. The Logit-vd model allows the probability

that a rating is missing to depend on both the value of the underlying rating and the identity of the item. The Logit-vd model specifies a logistic form for this relationship (hence the name Logit-vd) as seen in Equation 14.

$$P(r_{nd} = 1|x_{nd} = v) = \mu_{vd} = \frac{1}{1 + \exp(-(\sigma_v + \omega_d))} \quad (14)$$

The σ_v factor models a non-random missing data effect that depends on the underlying rating value. This effect is constrained to be the same across all items. The ω_d factor models a per-item missing data effect. This effect can be useful if all items do not have the same exposure in a recommender system. This situation can arise, for example, when some items are more heavily promoted than others. A Bayesian network representation of the combined finite multinomial mixture model/Logit-vd model is given in Figure 1(c).

In the Logit-vd model, the response indicators are assumed to be independently sampled for each item, yielding the missing data model given in Equation 15. The model parameters σ_v and ω_d are given Gaussian prior distributions as seen in Equation 16. The specification of the underlying multinomial mixture model for complete data is again given by Equations 8 to 11.

$$P(\mathbf{r}_n|\mathbf{x}_n, \mu) = \prod_{d=1}^{D} \prod_{v=1}^{V} (\mu_{vd}^{[r_{nd}=1]}(1 - \mu_{vd})^{[r_{nd}=0]})^{[x_{nd}=v]} \quad (15)$$

$$P(\sigma, \omega|\xi, \nu) = \prod_{v=1}^{V} \mathcal{N}(\sigma_v|0, \xi_v^2) \cdot \prod_{d=1}^{D} \mathcal{N}(\omega_d|0, \nu_d^2) \quad (16)$$

3.4 Logit-vd and CPT-v Model Estimation

We present a generalized Expectation Maximization (EM) algorithm to simultaneously estimate the parameters of the Logit-vd missing data model and the multinomial mixture complete data model [3]. To help simplify the notation, we introduce the auxiliary variables γ_{dkn} in Equation 17.

$$\gamma_{dkn} = (\beta_{x_{nd}dk}\mu_{x_{nd}d})^{[r_{nd}=1]} \left(\sum_{v=1}^{V} \beta_{vdk}(1 - \mu_{vd})\right)^{[r_{nd}=0]} \quad (17)$$

The EM updates for the combined multinomial mixture/Logit-vd model are given in Algorithm 1. The σ_v and ω_d parameters are updated using a gradient step with the step length λ set using a line search on each iteration to ensure convergence.

The EM algorithm for the multinomial mixture/CPT-v model parameters follows directly from the Logit-vd case if we set $\mu_{vd} = \mu_v$ for all d in Equation 17, and replace the M-Step updates for σ_v and ω_d given in Algorithm 1 with the closed-form M-Step update given below.

$$C_{v1} \leftarrow \sum_{n=1}^{N} \sum_{d=1}^{D} [r_{nd} = 1][x_{nd} = v] \quad (18)$$

$$C_{v0} \leftarrow \sum_{n=1}^{N} \sum_{d=1}^{D} q_n(v, d)[r_{nd} = 0] \quad (19)$$

$$\mu_v \leftarrow \frac{\xi_{1v} - 1 + C_{v1}}{\xi_{1v} + \xi_{0v} - 2 + C_{v1} + C_{v0}} \quad (20)$$

Algorithm 1 Generalized EM for MM/Logit-vd

E-Step:

$$q_n(k) \leftarrow \frac{\theta_k \prod_{d=1}^{D} \gamma_{dkn}}{\sum_{k=1}^{K} \theta_k \prod_{d=1}^{D} \gamma_{dkn}}$$

$$q_n(k, v, d) \leftarrow q_n(k) \left(\frac{(1 - \mu_{vd})\beta_{vdk}}{\sum_w (1 - \mu_{wd})\beta_{wdk}}\right)^{[r_{nd}=0]}$$

$$q_n(v, d) \leftarrow \sum_{k=1}^{K} q_n(v, d)$$

M-Step:

$$\theta_k \leftarrow \frac{\alpha_k - 1 + \sum_{n=1}^{N} q_n(k)}{N - K + \sum_{k=1}^{K} \alpha_k}$$

$$C_{vdk} = \sum_{n=1}^{N} q_n(k)[r_{nd} = 1][x_{nd} = v] + q_n(k, v, d)[r_{nd} = 0]$$

$$\beta_{vdk} \leftarrow \frac{\phi_{vdk} - 1 + C_{vdk}}{\sum_{n=1}^{N} q_n(k) - V + \sum_{v=1}^{V} \phi_{vdk}}$$

$$\sigma_v \leftarrow \sigma_v - \lambda \sum_{d=1}^{D} \sum_{n=1}^{N} \frac{\partial E[\log \mathcal{P}]}{\partial \mu_{vd}} \mu_{vd}(1 - \mu_{vd}) - \frac{1}{\xi_v^2}(\sigma_v)$$

$$\omega_d \leftarrow \omega_d - \lambda \sum_{v=1}^{V} \sum_{n=1}^{N} \frac{\partial E[\log \mathcal{P}]}{\partial \mu_{vd}} \mu_{vd}(1 - \mu_{vd}) - \frac{1}{\nu_d}(\omega_d)$$

$$\mu_{vd} \leftarrow \frac{1}{1 + \exp(-(\sigma_v + \omega_d))}$$

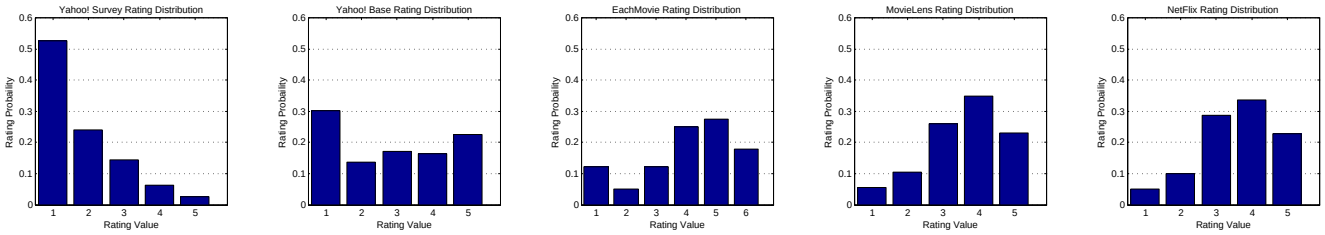
We note that learning the parameters of the multinomial mixture model under the missing at random assumption is accomplished using the standard EM algorithm for discrete mixtures. Further details for all models can be found in [7].

3.5 Logit-vd and CPT-v Discussion

CPT-v and Logit-vd employ simple conditional Bernoulli selection models for the response variables. The advantage of this form of missing data model is computational tractability. Pre-computing and caching intermediate factors gives highly efficient EM algorithms for both CPT-v and Logit-vd with approximately the same computational cost per iteration as learning the multinomial mixture under the missing at random assumption. The computational complexity in both cases depends on the number of observations, not the data matrix size. Both models of course ignore important information that might influence whether or not particular items will be observed, and do not incorporate feature-based information about users and items that could be very helpful in overcoming the effects of non-random missing data. Neither type of information is available for the data set we consider, but an advantage of a probabilistic approach is that the basic models can easily be extended to deal with additional features and side information.

3.6 Matrix factorization

We consider a probabilistic matrix factorization model with global mean offset as seen in Equations 21 to 23 [10]. U_n denotes a length K user factor vector while Y_d denotes a length K item factor vector. μ_g is the global average rating. The item parameter vector Y_d can be thought of as



(a) Yahoo! Random (b) Yahoo! User (c) EachMovie (d) MovieLens (e) NetFlix

Figure 2: Distribution of rating values for randomly selected items (Yahoo! Random) and user-selected items (Yahoo! User) compared to several popular collaborative filtering data sets including EachMovie, MovieLens, and Netflix.

representing the strength of a set of features describing each item. The user parameter vector U_d can be thought of as representing the user's affinity for items described by each feature. Training for the matrix factorization model is accomplished by numerical optimization of the log likelihood function. Without loss of generality we can assume $\sigma^2 = 1$, at which point learning the model is equivalent to standard regularized matrix factorization with penalty $\lambda = 1/\sigma_0^2$.

$$P(x_{nd}) = \mathcal{N}(x_{nd}|\mu_g + U_n Y_d^T, \sigma^2) \quad (21)$$

$$P(U_n) = \mathcal{N}(U_n|0, \sigma_0^2) \quad (22)$$

$$P(V_d) = \mathcal{N}(Y_d|0, \sigma_0^2) \quad (23)$$

3.7 Item-Based Nearest Neighbour Regression

The final method we consider is item-based nearest neighbour regression. We use an adjusted cosine similarity metric as shown below [12], combined with the standard nearest neighbour regression prediction rule. We found that restricting the prediction rule to positive similarities only resulted in better prediction performance as noted previously by Takács et al. [13]. $\bar{x}_n$ denotes the average of user n's observed ratings while $\hat{x}_{nd}$ denote the prediction for user n and item d.

$$W_{dd'} = \frac{\sum_{n=1}^{N} r_{nd} r_{nd'} (x_{nd} - \bar{x}_n)(x_{nd'} - \bar{x}_n)}{\sqrt{\sum_{n=1}^{N} r_{nd} (x_{nd} - \bar{x}_n)^2} \sqrt{\sum_{n=1}^{N} r_{nd'} (x_{nd'} - \bar{x}_n)^2}} \quad (24)$$

$$\hat{x}_{nd} = \frac{\sum_{d' \neq d} W_{dd'} r_{nd'} x_{nd'}}{\sum_{d' \neq d} |W_{dd'}| r_{nd'}} \quad (25)$$

4. THE YAHOO! DATA SET

Our empirical analysis is based on the *Yahoo! Music ratings for User-Selected and Randomly Selected Songs, version 1.0* data set, which is available through the Yahoo! Webscope data sharing program.[1] This data set is essentially identical to the data set used in our previous study [8]. It presents a unique opportunity to test collaborative filtering methods that incorporate missing data models. It contains ratings for items selected at random, in addition to ratings for items selected by the user. It permits the evaluation of rating prediction and ranking methods using a novel protocol that considers ratings for randomly selected items as described in Section 5.

The data set consists of ratings collected during normal user interaction with Yahoo's LaunchCast internet radio service, as well as ratings for items collected using an online survey. We will refer to the set of ratings collected through normal interaction with the recommender system as *ratings for user-selected items* and denote this set by X^u. We will refer to the set of ratings collected through the survey as *ratings for randomly selected items* and denote this set by X^r.

The data set contains $15,400$ users, all with at least 10 ratings for user-selected items in X^u. $5,400$ of these users also have exactly 10 ratings for randomly selected items in the set X^r. The data set is based on $1,000$ songs selected at random from the LaunchCast catalog. There are a total of approximately $250,000$ ratings for user-selected items in X^u and exactly $54,000$ ratings for randomly selected items in X^s.

Figure 2 shows the marginal distribution of ratings for randomly selected items in X^r compared to the distribution of ratings for user-selected items in X^u, and several other popular collaborative filtering data sets. The two sets of ratings X^u and X^r in the Yahoo! data set exhibit markedly different marginal statistics. The most pronounced feature of the ratings for randomly selected items in the Yahoo! data set is that they contain many fewer high ratings compared to the ratings for user-selected items. This points strongly to a possible violation of the missing at random condition in the Yahoo! ratings for user-selected items. Marlin et al. present further properties of the data set and additional arguments supporting a possible violation of the missing at random assumption in this data set [8].

5. EXPERIMENTAL PROTOCOL

The experimental protocol we employ is significantly more involved than typical collaborative filtering evaluation procedures based on historical rating data as we deal with two rating sources (user-selected and randomly selected items). In this section we describe in detail the data set manipulation needed to enable this experimental protocol, but the basic idea is very simple: we train models on ratings for user-selected items and test both on held-out ratings for user-selected items and held-out ratings for randomly selected items. We argue that testing on randomly selected items is a more accurate measure of performance than testing on items selected by the user since it better reflects an

[1]Contact academicrelations@yahoo-inc.com or visit http://research.yahoo.com/Academic_Relations for details on obtaining Yahoo! Webscope data sets.

important goal of recommender systems: to make predictions and recommend items that the user has not yet seen or rated.

5.1 Data Set Preparation

We begin by filtering the data so that each user has at least 11 ratings in the set X^u instead of the original 10. We choose 10 user-selected items from X^u to form a set of held-out user-selected test items X^{hu}. The remaining ratings in X^u form the set of user-selected observed ratings X^{ou}. The additional filtering insures at least one rating for each user in X^{ou}. All of the ratings for randomly selected items X^r are held-out for testing.

To enable a cross-validation assessment of weak/strong generalization error, we split the users with ratings for randomly selected items (the users who participated in the online survey and data collection experiment) into five equal sized groups. For each cross-validation fold, we select one of the five groups of users to form a set of test users. All of the remaining users form the set of training users. We obtain a total of six sets of ratings for each cross-validation fold: X^{ou}_{tr},X^{ou}_{te},X^{hu}_{tr},X^{hu}_{te},X^{r}_{tr}, and X^{r}_{te}.

We train models on the observed ratings for user-selected items contained in the training user set X^{ou}_{tr}. Conditioning on each training user's observed ratings in X^{ou}_{tr}, we evaluate weak generalization performance on held-out ratings for user-selected items contained in the training user set X^{hu}_{tr}. We separately evaluate weak generalization performance on ratings for randomly selected items contained in the training user set X^{r}_{tr}. Next, conditioning on each test user's observed ratings in X^{ou}_{te}, we evaluate strong generalization performance on held-out ratings for user-selected items contained in the test user set X^{hu}_{te}. We separately evaluate strong generalization performance on ratings for randomly selected items contained in the test user set X^{r}_{te}.

5.2 Hyper-Parameter Settings and Optimization Details

In the experiments that follow, we train each mixture model using $1, 5, 10$ and 20 mixture components. The Logit-vd Normal prior parameters ξ^2 and ν^2 were set to 10 to provide a broad prior around zero. The mixture model hyper-parameters α and ϕ and the CPT-v prior parameters ξ were all fixed to 2 to provide minimal smoothing. No attempt was made to update the hyper-parameters in this work. For the matrix factorization model, we trained for $10,000$ iterations using limited memory BFGS [9, p. 224] or until the change in the objective function was less than 10^{-5}. We considered ranks $K = 1, 5, 10, 20$, and regularization parameters $0.1, 1, 5, 10$. Each of the mixture models was trained for $10,000$ EM iterations or until the change in the average log posterior probability was less than 10^{-8}.

6. EVALUATION METRICS

We evaluate rating prediction performance in terms of normalized mean absolute error (NMAE), computed as seen below, assuming there are T test items per user with indices $i(1,n)$ to $i(T,n)$. The normalizing constant (equal to 1.6 here) is the expected MAE assuming uniformly distributed predictions and true ratings. When evaluating rating prediction performance, we set the predicted rating for item d and user n to the median of the posterior predictive distribution for that user/item combination since this minimizes the MAE in expectation. We truncate predictions to $[1, 5]$ when predicting ratings if the prediction method does not guarantee this property automatically.

$$NMAE = \sum_{n=1}^{N} \sum_{t=1}^{T} \frac{|x_{i(t,n)n} - \hat{x}_{i(t,n)n}|}{1.6NT}$$

$$NDCG@L = \sum_{n=1}^{N} \frac{\sum_{l=1}^{L} (2^{x_{\hat{\pi}(l,n)n}} - 1)/\log(1+l)}{N \sum_{l=1}^{L} (2^{x_{\pi(l,n)n}} - 1)/\log(1+l)}$$

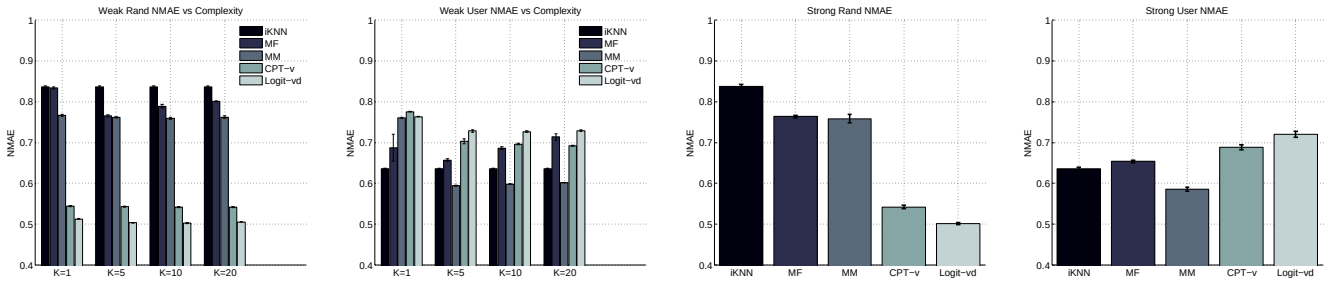
We evaluate ranking performance using a standard metric in information retrieval ranking applications, the normalized discounted cumulative gain (NDCG@L). The NDCG@L score is computed as seen above where $\pi(l,n)$ is the index of the item with rank l when test items are sorted in descending order by true rating x_{nd}, $\hat{\pi}(l,n)$ is the index of the item with rank l when items are sorted in descending order according to their predicted ratings $\hat{x}_{nd}$. When sorting by true and predicted ratings, ties can be broken arbitrarily without affecting the NDCG@L score. L denotes the length of the ranked list of items we return to the user. When evaluating ranking performance, we set the predicted rating for item d and user n to the mean of the posterior predictive distribution for that user/item combination. Note that *lower* NAME indicates better prediction performance while *higher* NDCG indicates better ranking performance.

7. RATING PREDICTION RESULTS

Figures 3(a) and 3(b) present the weak generalization rating prediction performance on randomly selected items and user-selected items for item-based nearest neighbour regression (iKNN), the matrix factorization model (MF), the multinomial mixture model (MM), the multinomial mixture/ CPT-v model combination (MM/CPT-v), and the multinomial mixture/Logit-vd model combination (MM/Logit-vd) as a function of the number of latent dimensions K. For MF the best values with respect to the regularization parameter λ are shown for each K. Note that we use all neighbours in the $iKNN$ method, so its performance is constant across K.

Figures 3(c) and 3(d) show the strong generalization performance on randomly selected items and user-selected items for each of the five models. Strong generalization results are reported for the complexity K giving the best weak generalization performance. The optimal model complexities were chosen independently for user-selected and randomly selected items. Note that the standard errors are represented on the plots using error bars.

The results on randomly selected test items clearly show that the MM/Logit-vd model achieves slightly better rating prediction performance than MM/CPT-v, while both achieve significantly better performance than the iKNN, MM and MF models that operate under the missing at random assumption. On user-selected items, the basic MM model significantly out-performs both MM/CPT-v and MM/Logit-vd. The best results we obtained with the MF and iKNN models are worse than for MM. This is likely due to the fact that the data is highly sparse and the MM model has the fewest parameters to learn ($4,000K$ for MM versus $16,000K$ for MF). More advanced forms of regularization may yield better performance for MF on user-selected items. The limited effect with respect to K for all methods is again likely also due to the relatively small data set size.



(a) Weak Rand NMAE (b) Weak User NMAE (c) Strong Rand NMAE (d) Strong User NMAE

Figure 3: Figures (a) and (b) present the weak generalization rating prediction performance on randomly selected items and user-selected items. Figures (c) and (d) present the strong generalization rating prediction performance on randomly selected and user-selected items. The results show that the MM/Logit-vd model achieves the best rating prediction performance on randomly selected items.

We note again that computing the prediction error on randomly selected test items is more relevant than prediction error on user-selected test items when the models are subsequently used in a score-and-sort ranking framework. Estimating prediction error on user-selected items only gives an estimate of prediction performance on items that were previously selected by the user. This is clearly an unreliable estimate of prediction performance over the whole rating matrix in the case of the Yahoo! data set, or the results on the two test sets would be equal.

8. RANKING RESULTS

The collaborative ranking task is to produce an ordered top-L list of highest rated items for each user. The NDCG@L score provides a measure of quality for top-L lists. Actually evaluating NDCG@L in the collaborative ranking case is complicated by the fact that we do not have access to the true ratings for all of the items each user has *not* rated. The standard procedure for estimating ranking performance is to use held-out lists of user-selected items. A unique feature of the Yahoo! data set is that we have access to an additional set of randomly selected test items. Due to the small overlap between randomly selected items and user-selected items when the Yahoo! data set was collected, the randomly selected items are a good approximation to a random sample of items *not* rated by each user.

We perform ranking experiments on the Yahoo! data set based on held-out lists of 10 randomly selected items and 10 user-selected items. Weak generalization NDCG@L estimates were computed for models with 1, 5, 10, and 20 latent dimensions. The optimal model complexity was determined independently for each value of the list length L based on weak generalization performance, and the corresponding strong generalization NDCG@L value is displayed. Figure 4(a) shows the strong generalization NDCG@L performance for each model estimated on lists of 10 randomly selected items, while Figure 4(b) shows the same comparison based on lists of 10 user-selected items. The results show that on the user-selected items, the MM model performs as well as MM/Logit-vd and MM/CPT-v, and all three outperform MF and iKNN. On lists of randomly selected items, MM/Logit-vd and MM/CPT-v both perform significantly better than the basic MM and MF models.

9. CONCLUSIONS AND FUTURE DIRECTIONS

We have presented new empirical results comparing the ranking and rating prediction performance of methods that assume the MAR condition and methods that include a model of the missing data mechanism. Results show that methods that include a non-random missing data model out-perform methods that assume the MAR condition on both the prediction and ranking tasks when the evaluation is based on randomly selected test items. We have argued that the use of randomly selected test items more accurately reflects the tasks of interest: prediction and ranking for items *not* previously rated by the user. A very interesting direction for future research is to consider combining methods that optimize ranking performance, as in the work of [14], while simultaneously accounting for the presence of non-random missing data.

10. ACKNOWLEDGMENTS

We would like to thank Sam Roweis and Malcolm Slaney for their important contributions to earlier stages of this research. This research was supported by Natural Sciences and Engineering Research Council of Canada and the Killam Trusts.

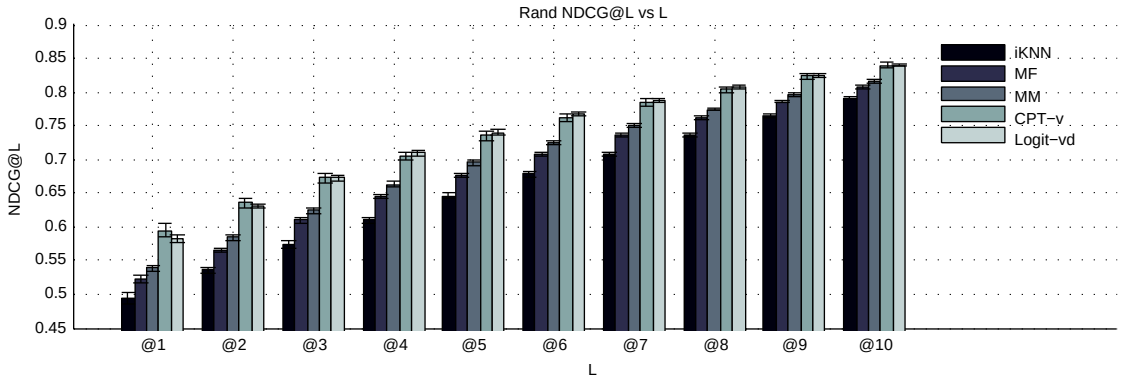
11. REFERENCES

[1] J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

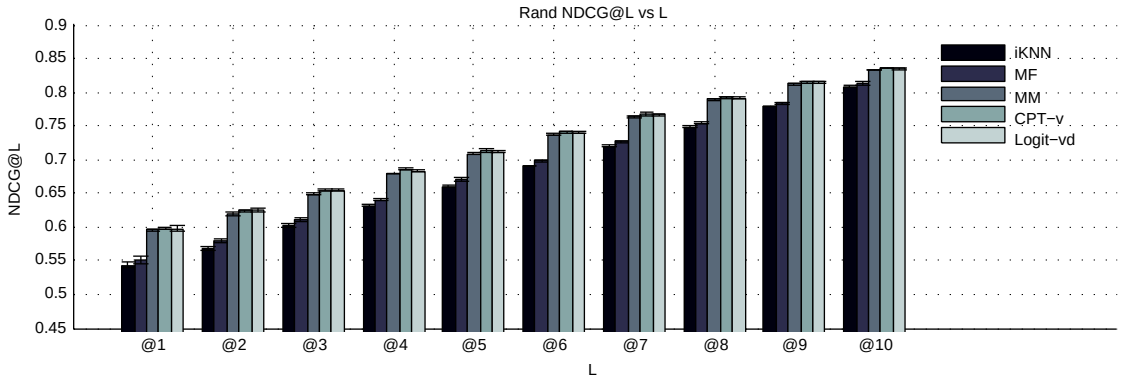
[2] D. Decoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 249–256, 2006.

[3] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[4] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd ACM SIGIR Conference*, pages 230–237, 1999.



(a) Strong Rand NDCG



(b) Strong User NDCG

Figure 4: Figures (a) and (b) compare NDCG@L estimates for iKNN, MF, MM, MM/CPT-v, and MM/Logit-vd on the Yahoo! data set.

[5] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250, 2000.

[6] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. John Wiley and Sons, Inc., 1987.

[7] B. Marlin. *Missing Data Problems in Machine Learning*. PhD thesis, University of Toronto, April 2008.

[8] B. Marlin, R. Zemel, S. Roweis, and M. Slaney. Collaborative filtering and the missing at random assumption. In *Uncertainty in Artificial Intelligence 23*, 2007.

[9] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[10] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[11] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, pages 249–256, 2007.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM.

[13] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Matrix factorization and neighbor based algorithms for the netflix prize problem. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 267–274, 2008.

[14] M. Weimer, A. Karatzoglou, Q. Le, and A. Smola. Cofi rank - maximum margin matrix factorization for collaborative ranking. In *Advances in Neural Information Processing Systems 20*, pages 1593–1600, 2008.

[15] M. Weimer, A. Karatzoglou, and A. Smola. Adaptive collaborative filtering. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 275–282, 2008.

[16] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130, 2008.

[17] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32, 2005.