The Multiple Multiplicative Factor Model For Collaborative Filtering

Benjamin Marlin MARLIN@CS.TORONTO.EDU
Richard S. Zemel ZEMEL@CS.TORONTO.EDU
Department of Computer Science, University of Toronto. Toronto, ON, M5S 3H5, CANADA

Abstract

We describe a class of causal, discrete latent variable models called Multiple Multiplicative Factor models (MMFs). The distinguishing feature of MMFs is that they combine component distributions multiplicatively, taking into account factor expression levels. The product formulation of MMFs allow factors to specialize to a subset of the items, while the causal generative semantics mean MMFs can readily accommodate missing data. This makes MMFs distinct from both directed mixture-type models and undirected product models. In this paper we present empirical results from the collaborative filtering domain showing that a binary/multinomial MMF model matches the performance of the best existing models while learning an interesting latent space description of the users.

1. Introduction

In this paper we introduce a class of directed latent variable models, Multiple Multiplicative Factor models (MMFs), which are applicable to data sets where multiple hidden factors may influence each data element. Figure 1 shows the graphical model for an MMF. The generative process for an MMF is as follows:

1. A discrete, non-negative expression level z_{nk} is selected independently for each of K factors Z_k.
2. Each element x_{nm} of data vector $\mathbf{x}_n$ is selected from a distribution proportional to the product of each factor's predicted distribution for element m, each raised to the power z_{nk}.

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

We obtain instances of the model relevant to particular types of data by making different assumptions about the domains and distributions of the latent factors and observed variables. If we assume the observed variables are continuous, and let the factor predictions represent parameters of Gaussian distributions, then the MMF model could be used to model gene expression data, for example. The MMF model can be applied to term-document data by assuming that each observed variable X_m represents the word in document position m, and that each factor encodes a different distribution over words. Similar assumptions allow the MMF model to be applied to the task of link prediction.

In the current work we present an application of the MMF model to the task of prediction in rating-based collaborative filtering. In the particular MMF model we apply, X_m is categorical and corresponds to a rating for item m. Each factor encodes a different multinomial distribution over rating values for each item. We assume the factors are binary valued and marginally independent.

The MMF model for collaborative filtering is quite different from other models that have been applied to rating prediction problems. The mixture of multinomials model, the aspect model (Hofmann, 2001), and the user rating profile model (Marlin, 2003) are all

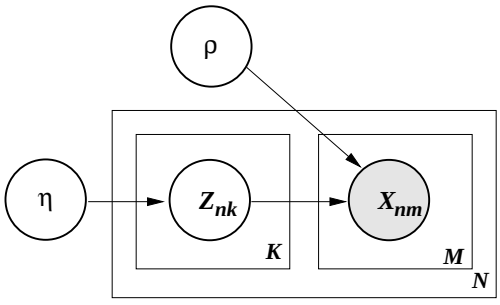

Figure 1. MMF Model

mixture-type models where component distributions are combined additively. Factor analysis and PCA are also linear, additive models (Canny, 2002). MMF does share one important similarity with these directed models: the ability to efficiently cope with any amount of missing data.

In this paper we describe the binary/multinomial multiple multiplicative factor model, present a variational learning procedure for MMF, give a brief overview of rating-based collaborative filtering, explain how the model is applied to the rating prediction task, present empirical results showing that MMF achieves state-of-the-art performance on two rating prediction data sets, and explore qualitative and quantitative properties of the learned representation.

2. The Binary/Multinomial MMF Model

In the binary/multinomial MMF model we assume the factor variables Z_k are binary valued and marginally independent. Each factor is Bernoulli distributed with mean η_k. The latent space description of each data vector $\mathbf{x}^n$ is a vector of factor expression levels $\mathbf{z}^n = (z_1^n, ..., z_k^n)$. Note that this vector does not represent a distribution so the factors Z_k can independently take on different expression levels z_k. The probability of a latent factor vector is:

$$P(\mathbf{Z} = \mathbf{z}|\eta) = \prod_{k=1}^{K} P(Z_k = z_k|\eta_k) \tag{1}$$

We assume each element x_m^n of each vector is categorical, taking one of V values. Each factor k contributes a multinomial distribution $P(X_m = v|k)$ over the values of X_m. We combine these distributions multiplicatively, taking into account the factor expression levels. Given a factor expression vector $\mathbf{z}^n$, we define the probability of observing value v for data element m as follows:

$$P(X_m = v|\mathbf{z}^n) = \frac{\prod_{k=1}^{K} P(X_m = v|k)^{z_k}}{\sum_{v'=1}^{V} \prod_{k=1}^{K} P(X_m = v'|k)^{z_k}} \tag{2}$$

While it is possible to parameterize the distributions $P(X_m = v|k)$ using basic multinomial parameters β_{vmk}, we avoid the use of constrained optimization in learning by instead using the natural parameters ρ_{vmk}. The natural parameters can take arbitrary real values and are related to the basic parameters through the softmax function: $\beta_{vmk} = \sigma(\rho_{vmk}) = \exp(\rho_{vmk}) / \sum_{v'} \exp(\rho_{v'mk})$. We can now re-express $P(X_m = v|\mathbf{z}^n)$ in terms of the natural parameters:

$$P(X_m = v|\mathbf{z}^n, \rho) = \frac{\exp(\sum_{k=1}^{K} z_k^n \rho_{vmk})}{\sum_{v'=1}^{V} \exp(\sum_{k=1}^{K} z_k^n \rho_{v'mk})} \tag{3}$$

In the collaborative filtering domain where we apply the model in this paper, we must make allowances for missing values in the data vectors. Here we treat missing values as missing at random. We define the function $\delta(x_m^n, v)$ to be 1 when $x_m^n = v$ and 0 otherwise. In particular it is 0 if x_m^n is unobserved. We define the variables $s_m^n = \sum_{v=1}^{V} \delta(x_m^n, v)$ to indicate if any value is observed for x_m^n. The probability of a data vector $\mathbf{x}^n$ given factor vector $\mathbf{z}^n$ is as follows:

$$P(\mathbf{x}^n|\mathbf{z}^n, \rho) = \prod_{m=1}^{M} \frac{\prod_{v=1}^{V} \exp(\sum_{k=1}^{K} z_k^n \rho_{vmk})^{\delta(x_m^n, v)}}{\sum_{v'=1}^{V} \exp(\sum_{k=1}^{K} z_k^n \rho_{v'mk})^{s_m^n}} \tag{4}$$

The multiplicative formulation of MMFs creates the possibility for cooperative factor learning, unlike in mixture models where factors are learned competitively. In particular, a factor k can have no opinion about a certain data element m by learning a set of ρ_{vmk} that are approximately uniform over the range of v. This may allow different factors to specialize to a subset of the data vector components, leading to a more efficient use of model parameters.

2.1. Variational Approximation

Exact inference in the binary/multinomial MMF model is impractical for large K due to a sum over all binary vectors of length K. However, exact inference can always be performed with binary valued factors in a finite amount of time. In order to develop a learning procedure, we employ a variational approximation to the true posterior $P(\mathbf{Z}|\mathbf{X} = \mathbf{x}^n)$. We assume a factorial Q-distribution for the hidden factor variables:

$$Q(\mathbf{Z} = \mathbf{z}|\mathbf{X} = \mathbf{x}^n, \mu) = \prod_{k=1}^{K} Q(Z_k = z_k|\mu_k^n) \tag{5}$$

The objective function of the binary/multinomial MMF model with respect to a single data case is given by the negative Kullback-Liebler divergence from the approximate to true posterior distribution over the latent variables: $F^n[\mu^n, \eta, \rho] = -KL(Q||P) = E_Q[\log P(\mathbf{x}^n, \mathbf{z}^n|\eta, \rho)] + H[Q(\mathbf{z}^n|\mathbf{x}^n, \mu^n)]$. Expanding this expression in terms of the specified distributions and parameters, and applying Jensen's inequality we obtain a tractable lower bound $\tilde{F}^n[\mu^n, \eta, \rho]$ on the per-vector objective function. Maximizing $\tilde{F}^n[\mu^n, \eta, \rho]$ corresponds to an approximate expectation maximization procedure (Neal & Hinton, 1998). We introduce the auxiliary variables $\gamma_{vmk}^n = \mu_k^n \exp(\rho_{vmk}) + 1 - \mu_k^n$ and $\alpha_{vm}^n = \prod_{k=1}^{K} \gamma_{vmk}^n$ to simplify the objective:

$$
\begin{aligned}
\tilde{F}^n[\mu^n,\eta,\rho] &= \sum_{m=1}^{M}\sum_{v=1}^{V}\delta(x_m^n,v)\sum_{k=1}^{K}\mu_k^u\rho_{vmk} \\
&- \sum_{m=1}^{M} s_m^n \log\sum_{v=1}^{V}\alpha_{vm}^n \\
&+ \sum_{k=1}^{K}(\mu_k^n\eta_k + (1-\mu_k^n)(1-\eta_k)) \\
&- \sum_{k=1}^{K}(\mu_k^n\log(\mu_k^n) + (1-\mu_k)\log(1-\mu_k^n))
\end{aligned}
$$

2.2. Learning

In this sub-section we describe a model fitting procedure for the binary/multinomial MMF model based on maximizing the objective function $\tilde{F}[\mu,\eta,\rho]$. Not surprisingly the gradient of $\tilde{F}[\mu,\eta,\rho]$ with respect to the μ_k^n parameters has the parameters coupled for each user, and the gradient of $\tilde{F}[\mu,\eta,\rho]$ with respect to ρ_{vmk} has all parameters coupled. Thus, we give formulas for computing these gradients, and describe nonlinear optimization procedures for variational inference and model fitting.

$$
\begin{aligned}
\frac{\partial\tilde{F}[\mu,\eta,\rho]}{\partial\mu_k^n} &= \sum_{m=1}^{M}\sum_{v=1}^{V}\delta(r_m^n,v)\rho_{vmk} \\
&- \sum_{m=1}^{M} s_m^n \sum_{v'=1}^{V}\lambda_m^n \frac{\exp(\rho_{vmk})-1)}{\mu_k^n(\exp(\rho_{vmk})-1)+1} \\
&+ \log(\eta_k) - \log(1-\eta_k) \\
&- \log(\mu_k^n) + \log(1-\mu_k^n) \qquad (6) \\
\frac{\partial\tilde{F}[\mu,\eta,\rho]}{\partial\rho_{vmk}} &= -\sum_{u=1}^{N} s_k^n\lambda_m^n \frac{\mu_k^n\exp(\rho_{vmk})}{\mu_k^n(\exp(\rho_{vmk})-1)+1} \\
&+ \sum_{n=1}^{N}\delta(r_m^n,v)\mu_k^n \qquad (7) \\
\frac{\partial\tilde{F}[\mu,\eta,\rho]}{\partial\eta_k} &= \eta_k - \frac{1}{N}\sum_{n=1}^{N}\mu_k^n \qquad (8) \\
\lambda_m^n &= \frac{\alpha_{vm}^n}{\sum_{v=1}^{V}\alpha_{vm}^n}
\end{aligned}
$$

Analytical updates for ρ_{vyk} and μ_k^u can not be found due to coupling of parameters in their respective gradient equations, hence we use iterative, nonlinear optimization techniques for learning. In the binary/multinomial MMF model the ρ_{vmk} parameters are unconstrained, but μ_k^n parameters represent Bernoulli probabilities and are constrained to lie within the interval $[0,1]$.

A number of optimization methods exist for iteratively solving box constrained optimization problems. However, since the number of users in a collaborative filtering data set ranges from tens of thousands to hundreds of thousands and the number of factor variables may be on the order of hundreds, clearly any method relying on second derivatives will be computationally intractable. Two methods that rely only on first order derivatives are the log-barrier method, and the projected gradient method (Bertsekas, 1982, p. 76). The log-barrier method is well known to exhibit extremely slow convergence in most cases. The projected gradient method is a modification of regular gradient descent. The method has a simple form for problems where each variable x_i is constrained to lie in the interval $[lb_i, ub_i]$. In this case the projected gradient method replaces the standard gradient descent step $x^{t+1} = x^t - \xi^t\nabla f(x^t)$ with the projected gradient step $x^{t+1} = \mathcal{P}(x^t - \xi\nabla f(x^t))$ where $\mathcal{P}(x)$ is the projection function. For box constrained problems

Algorithm 1 BinMMF-VarInf

Input: $\mathbf{x}$, η, ρ, I
Output: μ

Initialize μ_k, $\xi \leftarrow 1$
for $t = 1$ to I **do**
  for $k = 1$ to K **do**
    $d_k \leftarrow \frac{\partial\tilde{F}[\mu,\eta,\rho]}{\partial\mu_k^n}$
    while $(\tilde{F}^n[\mathcal{P}(\mu^n - \xi d),\eta,\rho] > \tilde{F}^n[\mu^n,\eta,\rho])$ **do**
      $\xi \leftarrow \kappa\xi$
    end while
    $\mu \leftarrow \mathcal{P}(\mu - \xi d)$
  end for
end for

Algorithm 2 BinMMF-Learn

Input: $\{\mathbf{r}^u\}$, K
Output: η, ρ

Initialize η, ρ. $\xi^0 \leftarrow 1$
while Not Converged **do**
  for $n = 1$ to N **do**
    $\mu^n \leftarrow$ BinMMF-VarInf$(\eta,\rho,\mathbf{x}^n,H(n))$
  end for
  for $v = 1$ to V, $m = 1$ to M, $k = 1$ to K **do**
    $d_{vmk} \leftarrow \frac{\partial\tilde{F}[\mu,\eta,\rho]}{\partial\rho_{vmk}}$
  end for
  while $(\tilde{F}[\mu,\eta,\rho - \xi d] > \tilde{F}[\mu^t,\eta,\rho])$ **do**
    $\xi \leftarrow \kappa\xi$
  end while
  $\rho \leftarrow \rho - \xi d$
  for $k = 1$ to K **do**
    $\eta_k \leftarrow \frac{1}{N}\sum_{n=1}^{N}\mu_k^n$
  end for
end while

$\mathcal{P}(x)_i = \text{median}(lb_i, x_i, ub_i)$ (Bertsekas, 1982, p. 92). To ensure convergence the step size α^t must be chosen by an inexact line search procedure which satisfies sufficient increase and curvature conditions. A backtracking line search is particularly easy to implement.

We obtain a variational inference procedure by iteratively maximizing $\tilde{F}^n[\mu, \eta, \rho]$ with respect to μ^n using the projected gradient method. It is important to note that while the μ^n parameters are coupled for each data case, they are not coupled *across* data cases. We summarize the resulting variational inference procedure in Algorithm 1. κ is a parameter that controls the speed of backtracking in the line search. Its value must satisfy $0 < \kappa < 1$.

An iterative procedure for learning the parameters ρ_{vmk} and η_k of the binary/multinomial MMF model can now be defined. The ρ_{vmk} parameters are unconstrained, so standard gradient descent with backtracking can be used. The η_k parameters have an analytic update. We give the model fitting procedure in Algorithm 2.

3. Collaborative Filtering

In rating-based collaborative filtering, users express their preferences by explicitly assigning ratings to items that they have accessed, viewed, or purchased. This form of data is becoming increasingly prevalent as many web sites offer the user the option of rating items such as movies, research papers, or even professors.

The principal information filtering task in collaborative filtering is commonly referred to as *recommendation*. In rating-based collaborative filtering this task has a natural decomposition into the task of *rating prediction*, and the task of computing recommendations from a set of predictions. Given a particular item and user profile, the goal of rating prediction is to predict the user's true rating for the item as accurately as possible. Recommendation can be accomplished by recommending the items with the highest predicted ratings.

The capability to predict ratings has other interesting applications. Rating predictions can be incorporated with content-based scores to create a preference augmented search procedure (Claypool et al., 1999). Rating predictions may also be used to tailor web sites or interfaces to a particular user's tastes.

A variety of rating prediction methods have been proposed for rating-based collaborative filtering. The original methods for this task, such as the GroupLens algorithm, were based on nearest neighbor regression using Pearson's correlation as a similarity measure (Resnick et al., 1994). Other techniques that have been applied to rating prediction construct explicit models. As we have already mentioned, probabilistic techniques include probabilistic principal components analysis (Canny, 2002), the aspect model (Hofmann, 2001), and the user rating profile model (Marlin, 2003). For a summary of these and other collaborative filtering models and methods see (Marlin, 2004).

4. Applying MMF to Collaborative Filtering

The binary/multinomial MMF model has an intuitive application to collaborative filtering. In the rating-based case, the data vectors $\mathbf{x}^n$ correspond to user rating profiles $\mathbf{r}^u$ where r^u_m is user u's rating for item m. The rating values are assumed to be ordinal, and on a scale from 1 to V. The latent factors Z_k have an interpretation as *user attitudes*. The latent space description of a user is thus a binary vector indicating which attitudes are expressed for that user. The multinomial distributions associated with each factor give that factor's distribution over rating values for each item.

The variational inference and learning procedures for the binary/multinomial MMF model can be applied in the collaborative filtering case without modification. To use the model for predicting missing rating values, all that remains is to derive prediction equations. We begin by applying the variational inference method to compute the variational parameters μ^a_k given the active user's rating profile $\mathbf{r}^a$. Computing the distribution $P(R_m|\mathbf{R} = \mathbf{r}^a)$ is impractical even when the variational approximation is used because it involves a sum over all binary vectors of length K. To overcome this problem, we compute an approximation to the true predictive distribution by sampling factor vectors according to their probability under the Q-distribution. The factors are marginally independent so sampling a complete factor vector reduces to independently sampling each factor. As we will show later, the factors tend to be sparse so that a relatively small number of samples leads to very good results. We give the rating prediction equations below, and a complete rating prediction method in Algorithm 3. In practice a small number of samples gives very good prediction results due to the fact that most users appear to express relatively few factors.

Algorithm 3 BinMMF-Predict

Input: $\mathbf{r}^a$, ρ
Output: $\hat{\mathbf{r}}^a$

$\mu \leftarrow$ BinMMF-VarInf$(\eta, \rho, \mathbf{r}^a, H(a))$
for $s = 1$ to S **do**
  Sample $\mathbf{z}^s \tilde{}$ Bernoulli(μ)
end for
for $m = 1$ to M **do**
  for $v = 1$ to V **do**
    Compute $P^s(R_m = v|\mathbf{r}^a)$
  end for
  $\hat{r}^a_m \leftarrow$ median $P^s(R_m|\mathbf{r}^a)$
end for

$$P(R_m = v|\mathbf{r}^a) \approx \frac{P^s(R_m = v|\mathbf{r}^a)}{\sum_{v'=1}^{V} P^s(R_m = v'|\mathbf{r}^a)}$$

$$P^s(R_m = v|\mathbf{r}^a) = \sum_{s=1}^{S} \left(\frac{\exp(\sum_{k=1}^{K} z_k^s \rho_{vmk})}{\sum_{v'=1}^{V} \exp(\sum_{k=1}^{K} z_k^s \rho_{vmk})} \prod_{k=1}^{K} (z_k^s \mu_k^a + (1 - z_k^s)(1 - \mu_k^a)) \right)$$

5. Empirical Evaluation

We perform an empirical evaluation of the binary/multinomial MMF model based on two collaborative filtering data sets. The EachMovie data set was collected by the Compaq Systems Research Center over an 18 month period beginning in 1997. The data set contains 72916 users, 1628 movies and 2811983 ratings. Ratings are on a scale from 1 to 6. The data set is 97.6% sparse. The MovieLens data set was collected through the ongoing MovieLens project, and is distributed by GroupLens Research at the University of Minnesota. MovieLens contains 6040 users, 3900 movies, and 1000209 ratings collected from users who joined the MovieLens recommendation service in 2000. Ratings are on a scale from 1 to 5. The base data set is 95.7% sparse.

We begin by filtering each data set to contain users that have rated at least 20 items. In the case of EachMovie, this leaves about 35000 users and 1600 items. Filtering the MovieLens data set leaves just over 6000 users and 3500 movies. Next, we form three random partitions of each data set into training and testing users. We form three random partitions of EachMovie into 30000 training users and 5000 test users, and three random partitions of MovieLens into 5000 training users and 1000 test users. We hold out the observed rating of one item for each user in the test set. Note that MMF was trained using a random subset of 5000 training users in the case of EachMovie.

We apply a *strong generalization* experimental protocol (Marlin, 2004, p. 14). We begin by training each method using a set of training users. We then evaluate the ability of the trained method to predict the held out ratings for each test set user. The strong generalization protocol measures the ability of the method to predict ratings given novel user profiles. Note that this protocol is different than the *weak generalization* protocol normally used for collaborative filtering where methods are tested on held out *training user* ratings (Breese et al., 1998). The strong generalization performance of a collaborative filtering method is thus a better estimate of its performance in an online recommendation setting.

We evaluate generalization in terms of prediction performance using a normalized mean absolute error measure (NMAE). The standard mean absolute error measure used in collaborative filtering is $MAE = \frac{1}{N} \sum_{u=1}^{N} |\hat{r}^u_{y^u} - r^u_{y^u}|$, assuming one held out rating per user profile. Since we will be experimenting with data sets having different numbers of rating values we normalize the mean absolute error, which enables comparison across data sets. We define our NMAE error measure to be $MAE/E[MAE]$ where $E[MAE]$ denotes the expected value of the MAE assuming uniformly distributed observed and predicted rating values. For EachMovie $E[MAE] \approx 1.9444$ and for MovieLens $E[MAE] = 1.6$.

We determine the strong generalization NMAE for each of the three sets of test users from a given base data set (EachMovie or MovieLens). We compute the mean NMAE across these three test sets as well as the standard error of the mean. For models with a size parameter K we repeat the evaluation procedure for several settings of this parameter.

6. Results

We present prediction performance results comparing the binary/multinomial MMF model with a range of other methods and models for rating prediction including a simple multinomial model (Multi), a Pearson's correlation best K neighbor method (PKNN), a multinomial mixture model (MixMulti), and the URP model (URP). The graphs in Figure 2 report the lowest mean NMAE rate attained by each method for methods with a model size parameter K. We report the model size below the name of each method. These re-

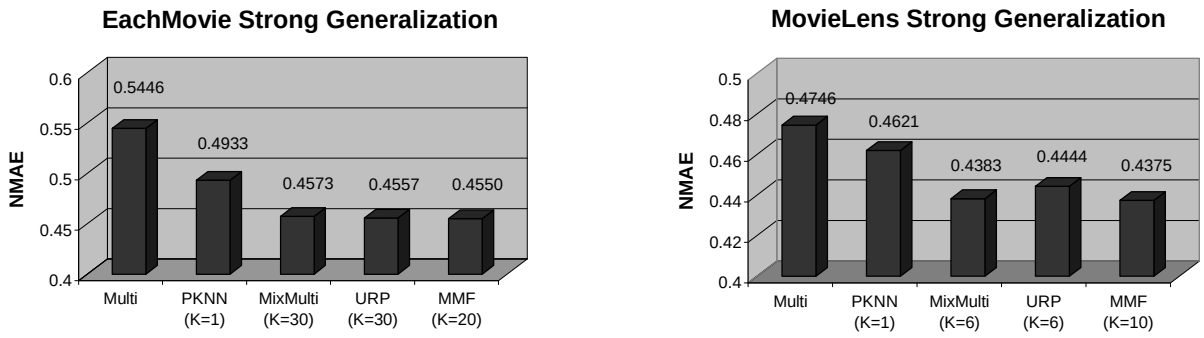

Figure 2. Prediction results for the multinomial model (Multi), Pearson's correlation K-nearest neighbour (PKNN), the multinomial mixture model (MixMulti), the user rating profile model (URP), and MMF on the EachMovie and MovieLens data sets. The differences in performance between MixMulti, URP and MMF are not statistically significant in either data set. The difference in performance between the best methods and both PKNN and Multi are statistically significant.

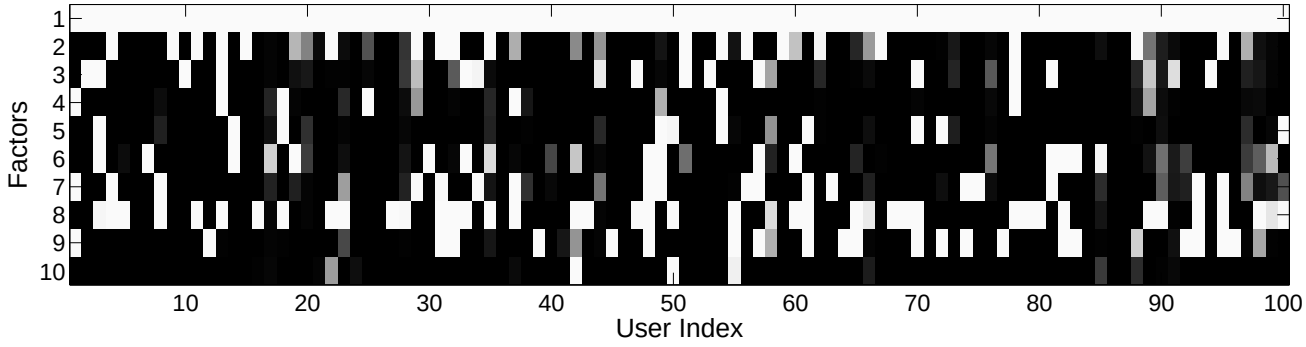

Figure 3. Factor activations are shown for a random set of 100 training users from the MovieLens data set. Black indicates the factor is on with probability 0, white indicates that a factor is on with probability 1. The first row represents a bias factor, which we clamp to 1 during learning for all users.

sults show that the binary/multinomial MMF rating prediction method attains strong generalization performance that is statistically equivalent to the most accurate currently known methods (URP and mixture of multinomials) on both the MovieLens and EachMovie data sets.

In addition to rating prediction performance, we study other qualitative and quantitative properties of the learned MMF model. All the results that follow were obtained using the binary/multinomial MMF model of size 10 which achieved the best rating prediction accuracy on the MovieLens data set.

In Figure 3 we show a representation of the variational parameters μ_k^u for a random subset of 100 training users. μ_k^u indicates the probability that factor k is activated for user u. By looking at the columns of Figure 3, we see that the number of activated factors for each user is quite low. By looking at the intensity of the cells, we see that factors tend to either be completely off, or completely on. Note that the first row of activations correspond to a bias factor, whose activation is clamped to 1 during learning. The fact that the model performs well while achieving sparse latent space descriptions is an appealing property. This justifies the use of a relatively low number of factor vector samples when computing predictions.

Each factor is represented as a multinomial distribution over ratings for each item. To gain insight into these distributions and how they combine to form the predictive distribution over rating values for a given user, we selected an item that represents a particularly hard case. From all items rated by greater than 1000 users, we selected the item with the highest entropy empirical distribution over rating values. Intuitively this is a hard case because equally large numbers of people have very different opinions about the item. Ideally, we would like the model to learn a representation where different factors place probability mass on different rating values. We show the set of multinomial distributions learned by the model in Figure 4a where the intensity of each cell is proportional to its probability. A close inspection reveals that factor 5

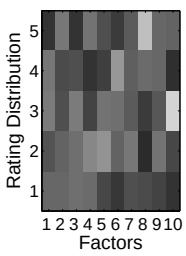

Figure 4a

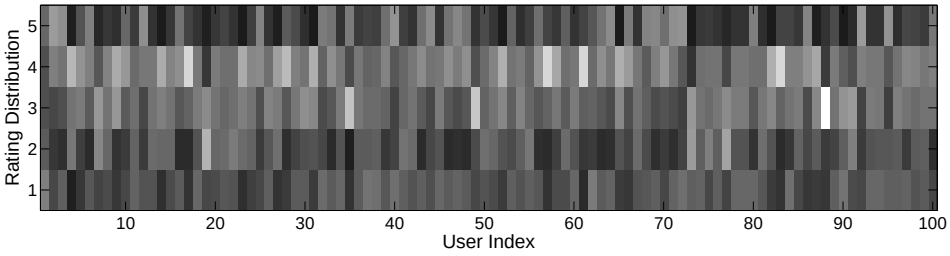

Figure 4b

Figure 4. The matrix at the left shows each factor's distribution over rating values for a single item. The matrix on the right shows the posterior distribution over ratings for the set of 100 users in Figure 3. The posterior distribution was computed using the variational approximation and sampling procedure given in Equation 9. A black cell represents a probability of 0, while a white cell represents a probability of 0.5.

has a clear maximum on value 2, factor 10 has a clear maximum on value 3, factor 6 has a clear maximum on value 4, and factor 8 has a clear maximum on value 5. On the other hand, factors 1, 2, and 3 are more uniform with no obvious single peak. These results are an indication of the type of cooperative learning and specialization that are possible with MMFs.

Next, we computed the posterior over rating values for the item in question using the factor activations of the users depicted in Figure 3. The posterior distribution was computed using the variational approximation and sampling procedure given in Equation 9. We show the results in Figure 4b. Several users exhibit sharper distributions over rating values than occur in the factors themselves. In particular, user 88 has a higher probability on rating value 3 than occurs in any of the factor distributions. This is a unique ability of a model based on combining distributions using products, instead of sums. A mixture model necessarily averages distributions creating a posterior distribution over ratings that is no sharper than any of the component distributions.

Another pertinent question about the binary/multinomial MMF model is the quality of the variational inference procedure. With 10 factors there are 1024 joint configurations of the factor vectors, and it is still possible to compute the exact distribution over all joint configurations of their settings given sufficient resources. We took advantage of this fact to compute the exact distribution over joint configurations of the latent variables for a random sample of 1000 users based on the 10 factor MMF model learned on the MovieLens data set. We then computed the variational approximation to the distribution over all joint configurations of the latent variables for these same users. Lastly, we computed the Kullback-Liebler divergence from the exact distribution to the variational approximation. We show a histogram of KL divergence values in

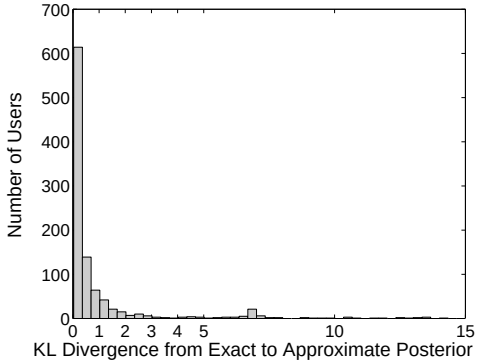

Figure 5. KL divergence from exact distribution over joint settings of the latent factors to the variational approximation for a trained 10 factor binary/multinomial MMF model.

Figure 5. It is clear from the histogram that for the majority of users in our random sample, the number of excess bits is less than one. This is quite good considering the granularity of the factorial variational distribution we have assumed.

7. Conclusions and Future Work

The key trait that distinguish of MMFs from previous directed latent variable models is the multiplicative combination of factor distributions. As we have seen, the product form has the advantage that it can create sharper distributions than the individual factors' distributions, allows factors to specialize to a subset of the data vector components, and mitigates some of the explaining-away problem that plagues most causal generative models.

MMFs can be considered as a directed analog to the undirected models defined by the product of experts (PoEs) (Hinton, 2002). PoEs have the advantage that inference is easy because the latent variables are con-

ditionally independent given the data, but MMF models have the advantage that they naturally handle any amount of missing data due to their directed structure.

The empirical results on the collaborative filtering task show that the binary/multinomial model matches the performance of the best known methods, while learning an interesting, sparse latent space description of the users. This apparent sparsity justifies the use of a relatively small number of factor vector samples during prediction. The results presented here were based on only 100 out of a possible 1024 samples.

The major drawback of MMFs is the complexity of learning and inference. Our variational approximation has proved to be effective on the tasks we have investigated so far, and the amount of computation required can be carefully controlled by imposing limits on the number of gradient ascent steps. While any scheduling of the updates improves the objective function, further testing is required to establish the tradeoff between limiting computation and the quality of the learned representation.

We are currently examining other instances of MMF models. First, we are exploring an Integer/Multinomial version, which will provide finer control when combining the factor predictions (see Equation 3). Here the flexibility in the factor expression levels trades off against more complicated constraints during inference. We plan to explore versions of the MMF model for other tasks such as text analysis, and gene expression analysis. In these offline domains, the computational complexity of learning and inference in the MMF model is less of an issue.

We are considering adding a level in the graphical model to encode interdependencies between the factors. For example, incorporating some topographic order in the latent space could improve representations and visualization in applications such as latent semantic analysis. Finally, an input-output version of the MMF would make it appropriate for classification problems, and has some interesting relationships to methods such as logarithmic opinion pools (Bordley, 1982; Heskes, 1998).

References

Bertsekas, D. P. (1982). *Constrained optimization and lagrange multiplier methods*. New York: Academic Press.

Bordley, R. (1982). A multiplicative formula for aggregating probability assessments. *Management Science*, *28*, 1137–1148.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence* (pp. 43–52).

Canny, J. (2002). Collaborative filtering with privacy via factor analysis. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 238–245).

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. *Proceedings of ACM SIGIR Workshop on Recommender Systems*.

Heskes, T. (1998). Selecting weighting factors in logarithmic opinion pools. *Proceedings of NIPS 10* (pp. 266–272).

Hinton, G. E. (2002). Training product of experts by minimizing contrastive divergence. *Neural Computation*, *14*, 1771–1800.

Hofmann, T. (2001). Learning What People (Don't) Want. *Proceedings of the European Conference on Machine Learning (ECML)*.

Marlin, B. (2003). Modeling user rating profiles for collaborative filtering. *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS-2003)*.

Marlin, B. (2004). Collaborative filtering: A machine learning perspective. Master's thesis, University of Toronto.

Neal, R. M., & Hinton, G. E. (1998). A new view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan (Ed.), *Learning in graphical models*, 355–368. Kluwer Academic Publishers.

Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., & Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work* (pp. 175–186). Chapel Hill, North Carolina: ACM.