A hierarchical model for ordinal matrix factorization

Ulrich Paquet ULRIPA@MICROSOFT.COM
Microsoft Research Cambridge
Cambridge CB3 0FB, United Kingdom

Blaise Thomson BRMT2@CAM.AC.UK
Engineering Department
University of Cambridge
Cambridge CB2 1PZ, United Kingdom

Ole Winther OWI@IMM.DTU.DK
Informatics and Mathematical Modelling
Technical University of Denmark
DK-2800 Lyngby, Denmark

Preprint. To appear in Statistics and Computing. The final publication is available at www.springerlink.com.

Abstract

This paper proposes a hierarchical probabilistic model for ordinal matrix factorization. Unlike previous approaches, we model the ordinal nature of the data and take a principled approach to incorporating priors for the hidden variables. Two algorithms are presented for inference, one based on Gibbs sampling and one based on variational Bayes. Importantly, these algorithms may be implemented in the factorization of very large matrices with missing entries. The model is evaluated on a collaborative filtering task, where users have rated a collection of movies and the system is asked to predict their ratings for other movies. The Netflix data set is used for evaluation, which consists of around 100 million ratings. Using root mean-squared error (RMSE) as an evaluation metric, results show that the suggested model outperforms alternative factorization techniques. Results also show how Gibbs sampling outperforms variational Bayes on this task, despite the large number of ratings and model parameters. Matlab implementations of the proposed algorithms are available from `cogsys.imm.dtu.dk/ordinalmatrixfactorization`.

Keywords: Large scale machine learning, collaborative filtering, ordinal regression, low rank matrix decomposition, hierarchial modelling, Bayesian inference, variational Bayes, Gibbs sampling

1. Introduction

Matrix factorization is a highly effective technique for both predictive and explanatory data modeling. An observed data set is represented as an $M \times N$ matrix matrix of results, $\mathbf{R}$, where rows represent the observation number and columns represent the variables of interest. The most common factorization is the bilinear decomposition $\mathbf{R} = \mathbf{U}^\top \mathbf{V} + \boldsymbol{\epsilon}$, where $\mathbf{U}$ is a $K \times M$ matrix, $\mathbf{V}$ is a $K \times N$ matrix, and $\boldsymbol{\epsilon}$ is a noise term. $\mathbf{U}$ and $\mathbf{V}$ represent the values of explanatory variables which, when multiplied and added, give a predictor of the values

in $\mathbf{R}$. If entries in $\mathbf{R}$ are missing, then $\mathbf{U}$ and $\mathbf{V}$ can be used to give predictions of their value. When $\mathbf{U}$ is observed the problem becomes one of multiple regression.

In many applications, the values of $\mathbf{R}$ will be further constrained. Although standard matrix factorization may still be used, an approach which explicitly models these constraints will often be more effective. In cases where the entries in $\mathbf{R}$, $\mathbf{U}$ and $\mathbf{V}$ are non-negative, non-negative matrix factorization algorithms have frequently improved performance. Examples of this improvement include tasks such as text mining and spectral analysis (Berry et al., 2007). This paper studies problems with a slightly different set of constraints. Entries in $\mathbf{R}$ are restricted to a finite ordered (ranked) set of values, and the task is therefore called ordinal matrix factorization.

Ordinal matrix factorization has several possible application areas. When the low-rank factors are constrained so that both rows and columns can simultaneously cluster – say when the entries in $\mathbf{R}$, $\mathbf{U}$ and $\mathbf{V}$ are all binary – matrix factorization provides a suitable framework for bi-clustering and pattern discovery of gene-expression data (Zhang et al., 2009, Shen et al., 2009). An example of the use of non-binary ordinal values in $\mathbf{R}$ is collaborative filtering. In this application, the rows of $\mathbf{R}$ correspond to items, its columns represent viewers or users, and its entries represent user ratings. $\mathbf{U}$ then provides the latent factors for a particular item, while $\mathbf{V}$ provides weights for how these factors affect a particular user's preferences.

Much of the relevant research on ordinal matrix factorization has been restricted to special cases. In binary matrix factorization, one suggested approach has been to constrain non-negative matrix factorization algorithms (Zhang et al., 2009, Shen et al., 2009). $\mathbf{U}$ and $\mathbf{V}$ are chosen to minimize the sum of squares error, $\sum_{ij} \|\mathbf{R}_{ij} - (\mathbf{U}^\top \mathbf{V})_{ij}\|^2$, subject to the constraints $\mathbf{U}_{ij}, \mathbf{V}_{ij} \in \{0, 1\}$. The extra constraints, however, result in the NP–hard discrete basis problem (Miettinen et al., 2008). This approach also lacks a simple extension to the more general case of ordinal matrix factorization. In the case of ordinal regression, which is when $\mathbf{U}$ and $\mathbf{R}$ are both observed, several algorithms have been suggested of which generalized linear models are one classic example (McCullagh and Nelder, 1989, Chu and Ghahramani, 2005). However, these algorithms have typically been limited to regression and have not been extended to the general factorization case.

There are two main approaches which have been applied to inference in the general ordinal matrix factorization task. The first is to define a loss function and optimise over $\mathbf{U}$ and $\mathbf{V}$ using maximum margin matrix factorization (Srebro et al., 2005). The second is to build a probabilistic model representing the matrix factorization and then to perform statistical inference to compute any desired values (Lim and Teh, 2007, Salakhutdinov and Mnih, 2008b, Stern et al., 2009).

This paper extends the latter approach by constructing a principled statistical model for ordinal matrix factorization, by combining the hierarchical model of Salakhutdinov and Mnih (2008a) with the ordinal regression likelihood of Chu and Ghahramani (2005). This is done through the inclusion of an additional *hidden* matrix $\mathbf{H} = \mathbf{U}^\top \mathbf{V} + \boldsymbol{\epsilon}$, which is then used as an unobserved input to an ordinal regression model to obtain $\mathbf{R}$. Contributions of the paper are:

- An extended probabilistic model for ordinal matrix factorization, in which non-linearities are introduced through the ordinal likelihood function. The hyperparameters express

a rich model, but allow the factors in $\mathbf{U}$ and $\mathbf{V}$ to remain coupled through shared hyper-priors.

- Two efficient algorithms, from different approaches to statistical inference, are derived and compared for the model. The first is a *Gibbs sampling* algorithm, which samples from the posterior distribution of all unobserved parameters. The second is a *variational* approach, in which the posterior distribution is approximated with a simpler surrogate distribution.

The model is tested on the Netflix data set, which is a collection of around 100 million movie ratings for $M = 17,770$ movie titles and $N = 480,189$ users. The matrix is sparse in that many of the movie-user pairs have no rating in the $\mathbf{R}$ matrix. Performance is computed by estimating missing values and computing the root mean-squared error (RMSE) on a held-out collection of ratings. The model is shown to outperform or be on a par with alternative matrix factorization approaches on this metric, which is the standard for evaluations on this particular data set. Note, however, that alternative approaches to collaborative filtering are available, such as factorizations that also regress against user and item features (Ansari et al., 2000, Stern et al., 2009), those using user rating profiles (Marlin, 2004), restricted Boltzmann machines (Salakhutdinov et al., 2007), nearest neighbours (Bell and Koren, 2007), and non-parameteric models (Lawrence and Urtasun, 2009, Yu et al., 2009a, Zhu et al., 2009). The best performance on this data has been achieved by averaging ensembles of many different models (Koren, 2009, Töscher et al., 2009, Piotte and Chabbert, 2009).

Besides the improvements obtained in RMSE, the model has other advantages over alternative matrix factorization techniques. Most importantly, the model is not restricted to providing only the expected value of missing entries in $\mathbf{R}$, and is able to provide a probability distribution over the possible discrete values. While RMSE cannot be improved using this extra information, other metrics such as the mean absolute error (MAE) would benefit from it. A further benefit of a Bayesian approach is that recommendations include a predictive variance. Practical systems can exploit this to only recommend items that it is *confident* a user will like.

The remainder of the paper is organized as follows: Section 2 describes the proposed hierarchical model, and its relationship with alternatives. Sections 3 and 4 then describe how efficient inference may be preformed using Gibbs sampling and variational Bayes, respectively. An evaluation of the framework is provided in Section 5, which describes the results on the Netflix data set. Directions for improving the core model are presented in Section 6.

2. Probabilistic model

The hierarchical Bayesian model for ordinal matrix factorization employed here comprises of three main elements: **1.** An ordinal regression likelihood function which maps a latent variable h to probabilities for the possible discrete ranked values r in such a way that the ordering is preserved; **2.** A probabilistic model for h in terms of a low rank matrix factorization; **3.** Hierarchical prior distributions for the low rank matrices. Each of these elements are discussed separately below, and combined in the graphical model presented in Figure 1.

2.1 Ordinal regression likelihood

Ordinal regression arises when the possible observed values in $\mathbf{R}$ are ranked. For example, in a collaborative filtering task an item with a five-star rating is regarded as superior to one with a four-star rating, which in turn is better than one with a three-star rating. In the true sense of the word there may be no definition of "distance" between ranked values; we can merely say that class A is preferred to class B, which is preferred to C, etc. (Stevens, 1946). Models for ordinal regression should therefore reflect both the discrete nature and natural ordering of the data. Such models hold an advantage over models which merely transform the problem into a regression problem, as the probability of a certain discrete ranked value can be computed.

Without loss of generality, the possible ranked values may be labeled $r = 1, \ldots, R$. For the remainder the paper, the different values are simply called ranks. The real line is now partitioned into a number of contiguous intervals with boundaries b_r,

$$-\infty = b_1 < b_2 < \ldots < b_{R+1} = +\infty \ ,$$

such that interval $[b_r, b_{r+1})$ corresponds to discrete rank r. Instead of directly modeling the ranks, the ranks are modeled in terms of a hidden variable f from a corresponding region of the real line. If f falls in rank r's interval, we know with full certainty that rank r is observed, and the conditional probability of r is therefore

$$p(r|f) = \begin{cases} 1 & \text{if } b_r \leq f < b_{r+1} \\ 0 & \text{otherwise} \end{cases} = \Theta(f - b_r) - \Theta(f - b_{r+1}) \ ,$$

where $\Theta(\cdot)$ is the step function, i.e. $\Theta(\cdot) = 1$ for a non-negative argument and zero otherwise.

Uncertainty about the exact location of f can be modelled by for example $p(f|h) = \mathcal{N}(f; h, 1)$. Averaging over f in $p(r, f|h) = p(r|f)p(f|h)$ gives

$$p(r|h) = \Phi(h - b_r) - \Phi(h - b_{r+1}) \ , \tag{1}$$

where $\Phi(x) = \int_{-\infty}^{x} \mathcal{N}(z; 0, 1)\, dz$ is the cumulative Gaussian density or probit function. When only two ranks are present, (1) becomes the familiar binary classification likelihood.[1] Another interpretation of this likelihood function is to view $p(r \geq r'|h) = \Phi(h - b_{r'})$, so that

$$p(r = r'|h) = p(r \geq r'|h) - p(r \geq r' + 1|h) \ .$$

Let r_{mn} denote the ratings for rows $m = 1, \ldots, M$ and columns $n = 1, \ldots, N$. The ordinal regression model maps continuous latent variables h_{mn} in $\mathbf{H}$ to probabilities $p(r_{nm}|h_{mn})$. The probability of observation r_{mn} can also be written as

$$p(r_{mn}|h_{mn}) = \prod_r \Big[\Phi(h_{mn} - b_r) - \Phi(h_{mn} - b_{r+1})\Big]^{\mathbb{I}[r_{mn}=r]} \ , \tag{2}$$

where the indicator function $\mathbb{I}[\cdot]$, which is one for a true argument and zero otherwise, picks out the $r = r_{mn}$ term. The likelihood defined over a training set of ranks $\mathcal{D} = \{r_{mn}|(m, n) \in$

1. Other sigmoid functions, like the logit $\sigma(x) = 1/(1 + e^{-x})$, can equally be chosen. The probit function is especially convenient for tractable inference, as will become evident in following sections.

tr.set} is

$$p(\mathcal{D}|\mathbf{H}) = \prod_{(m,n)} p(r_{mn}|h_{mn}) \ ,$$

where the product is over all observed ratings (m, n).

In this work the boundaries b_r are kept fixed and not included in the above likelihood. However, one may extend the parameter space to, for example, include a set of boundaries $\mathbf{b}_m$ for each item m. The b_r terms could also be learned from data, as they will contain information about the relative frequency in which ratings will appear.

2.2 Low rank matrix factorization

Collaborative filtering is possible if the latent correlations between items and users can be captured in h_{mn}. One choice towards this aim is to model h_{mn} with a linear model

$$h_{mn} = \mathbf{u}_m^\top \mathbf{v}_n + \epsilon_{mn} \ ,$$

with ϵ_{mn} being the residual and $\mathbf{u}_m$ and $\mathbf{v}_n$ being factors of length K associated with item m and user n (Hofmann, 1999, Rennie and Srebro, 2005). When K is smaller than M and N the linear model becomes a low rank decomposition $\mathbf{H} = \mathbf{U}^\top \mathbf{V} + \boldsymbol{\epsilon}$ in matrix form. Predictions for test cases (m', n') also rely on the dot product $\mathbf{u}_{m'}^\top \mathbf{v}_{n'}$ in low rank decomposition models, and this is equally true in the Bayesian approach adopted in this work.

A computationally convenient choice for modelling ϵ_{mn} is as a Gaussian random variable $\epsilon_{mn} \sim \mathcal{N}(0, \gamma^{-1})$, so that the prior distributions for all the latent variables share a variance γ^{-1} with

$$p(h_{mn}|\mathbf{u}_m, \mathbf{v}_n, \gamma) = \mathcal{N}(h_{mn}\,;\, \mathbf{u}_m^\top \mathbf{v}_n,\, \gamma^{-1}) \ . \tag{3}$$

As will be seen in Section 5, the choice of γ and K plays a key role in the performance of the model.

2.3 Hierarchical Bayesian priors

A standard Bayesian approach to modeling uncertainty in model parameters $\mathbf{U}$, $\mathbf{V}$, and γ, is by specifying hierarchical priors for each. If the priors are in the exponential conjugate family of distributions for each parameter, each of the conditional distributions required for Gibbs sampling is analytically tractable, and the updates needed for variational Bayesian inference follow a similar form. When enough data observations are present, inference is largely robust with respect to the actual choice of hyperparameters that govern the hierarchical priors, as the data likelihood typically dominates the prior.

Let each of the M item factors have a Normal distribution

$$p(\mathbf{u}_m|\boldsymbol{\mu}_u, \boldsymbol{\Psi}_u) = \mathcal{N}(\mathbf{u}_m; \boldsymbol{\mu}_u, \boldsymbol{\Psi}_u^{-1}) \ , \tag{4}$$

so that they are conditionally independent given the shared mean and covariance. The mean and precision matrix (inverse covariance) is modelled with a conjugate Normal-Wishart prior

$$p(\boldsymbol{\mu}_u, \boldsymbol{\Psi}_u) = \mathcal{N}(\boldsymbol{\mu}_u; \boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Psi}_u)^{-1})\, \mathcal{W}(\boldsymbol{\Psi}_u; \mathbf{W}_0, \nu_0) \ . \tag{5}$$

The Wishart distribution

$$\mathcal{W}(\mathbf{\Psi}; \mathbf{W}, \nu) \propto |\mathbf{\Psi}|^{(\nu-K-1)/2} \exp(-\frac{1}{2}\mathrm{tr}[\mathbf{W}^{-1}\mathbf{\Psi}])$$

over symmetric positive definite matrices is parameterized with a scale matrix $\mathbf{W}$ and ν degrees of freedom. A completely analogous model is used for the user factors. The hierarchical prior structure on $\mathbf{U}$ and $\mathbf{V}$, as specified here, is the same as that used in the Bayesian probabilistic matrix factorization model of Salakhutdinov and Mnih (2008a).

The inverse variance parameter γ is non-negative and is modelled with its conjugate Gamma distribution

$$p(\gamma; a_0, b_0) = \Gamma(\gamma; a_0, b_0) \ ,$$

where $\Gamma(\gamma; a, b) \propto \gamma^{a-1} \exp(-\gamma/b)$. The hyperparameters $\{\beta_0, \mathbf{W}_0, \nu_0\}$ and $\{a_0, b_0\}$ have to be specified by the user.

Model summary. The joint distribution of data and model parameters

$$\theta = \{\mathbf{H}, \mathbf{U}, \mathbf{V}, \gamma, \boldsymbol{\mu}_u, \mathbf{\Psi}_u, \boldsymbol{\mu}_v, \mathbf{\Psi}_v\} \ ,$$

using the definitions in Equations (2) to (5), is

$$\begin{aligned} p(\mathcal{D}|\theta)p(\theta) = \prod_{(m,n)} p(r_{mn}|h_{mn}) \ p(h_{mn}|\mathbf{u}_m, \mathbf{v}_n, \gamma) \cdots \\ \prod_m p(\mathbf{u}_m|\boldsymbol{\mu}_u, \mathbf{\Psi}_u) \cdot \prod_n p(\mathbf{v}_n|\boldsymbol{\mu}_v, \mathbf{\Psi}_v) \cdots \\ p(\boldsymbol{\mu}_u, \mathbf{\Psi}_u) \, p(\boldsymbol{\mu}_v, \mathbf{\Psi}_v) \, p(\gamma) \ . \end{aligned}$$

Predictive distribution. If each of the possible observed values, r, are arbitrary labels, say A, B, and C, the probability $p(r|\mathcal{D})$ obtained in the model gives the probability for each possible output label. In the case when the actual observed value is numeric, it may be preferable to obtain other statistics as well. The point estimate $\hat{r}$ that minimizes the expected squared error $E(\hat{r}) = (\hat{r} - r_{\mathrm{true}})^2$, for instance, is the expected prediction for r,

$$\hat{r} = \langle r \rangle = \sum_{r=1}^{R} r \, p(r|\mathcal{D}) = \sum_{r=1}^{R} r \int p(r|\theta) \, p(\theta|\mathcal{D}) \, d\theta \ . \tag{6}$$

Since the evaluation of Section 5 uses a RMSE, the above average r is used in that section. When a model's performance is judged by the absolute error $E(\hat{r}) = |\hat{r} - r_{\mathrm{true}}|$, the optimal point estimate $\hat{r}$ would be the median of $p(r|\mathcal{D})$.

Using the likelihood in (1), the summation over r for constant θ parameters

$$\begin{aligned} \sum_{r=1}^{R} r \, p(r|\theta) &= \sum_{r=1}^{R} \Phi(h_{mn} - b_r) - R \, \Phi(h_{mn} - b_{R+1}) \\ &= \sum_{r=1}^{R} \Phi(h_{mn} - b_r) \end{aligned}$$

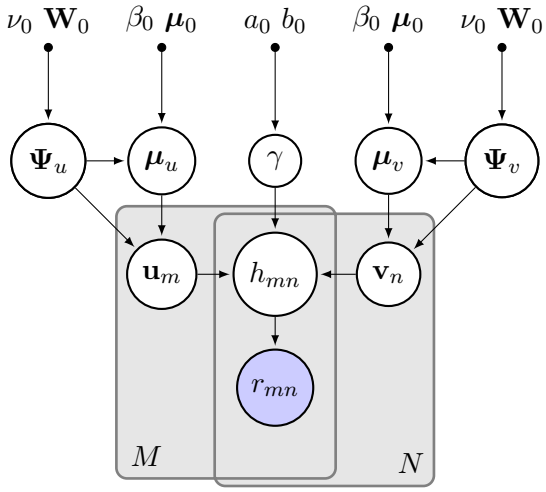

Figure 1: Graphical model for factor model Bayesian hierarchy as it is described in the main text.

shows that only the marginal distribution $p(h_{mn}|\mathcal{D})$ is needed to make predictions. We can further integrate h_{mn} out explicitly to show that

$$\sum_{r=1}^{R} r\, p(r|\mathbf{u}_m, \mathbf{v}_n, \gamma) = \sum_{r=1}^{R} \Phi\left(\frac{\mathbf{u}_m^\top \mathbf{v}_n - b_r}{\sqrt{1+\gamma^{-1}}}\right) . \tag{7}$$

This expression is then averaged over $p(\mathbf{u}_m, \mathbf{v}_n, \gamma|\mathcal{D})$ to make predictions. The following sections give two approaches to computing this average, either by sampling from the posterior parameter density with Gibbs sampling, or by approximating the parameter posterior by a "simpler" distribution that allows the (approximate) average to be analytically tractable.

Model interpretation. In (7) it is shown that the rating probability depends on the dot product $\mathbf{u}_m^\top \mathbf{v}_n - b_r$ in $\Phi(\cdot)$. The dot product allows the model to be interpreted as a linear classification problem over a binary matrix when $R = 2$, with similar interpretations for larger R or other likelihood functions. Each item factor $\mathbf{u}_m$ can be viewed as a classifier's "weight vector". Classifier m then classifies the user factors as "data points" $\mathbf{v}_n$ to its associated class rankings r_{mn}. Figure 2 illustrates the binary matrix case, with $\mathbf{b} = (-\infty, b_2, +\infty)$ and $b_2 = 0$.

In the model both the M weight vectors $\mathbf{u}_m$ and the N data points $\mathbf{v}_n$ are unobserved and can therefore be freely placed (according to their prior) in $\mathbb{R}^K$ to give high probability to the data point labels. If $\Pi(n)$ denotes the set of items rated by user n, then each $\mathbf{v}_n$ is associated with multiple labels r_{mn} for $m \in \Pi(n)$. The sets of labels $\Pi(n)$ therefore co-constrain and correlate the placement of $\mathbf{u}_m$ and $\mathbf{v}_n$, giving rise to their posterior distribution.

The interpretation presented here has a dual view where there are N classifiers with weight vectors $\mathbf{v}_n$, and M data points $\mathbf{u}_m$. Each point $\mathbf{u}_m$ is then associated with a set of ordinal labels r_{mn} for all $n \in \Omega(m)$, where $\Omega(m)$ indexes the users that rated item m.

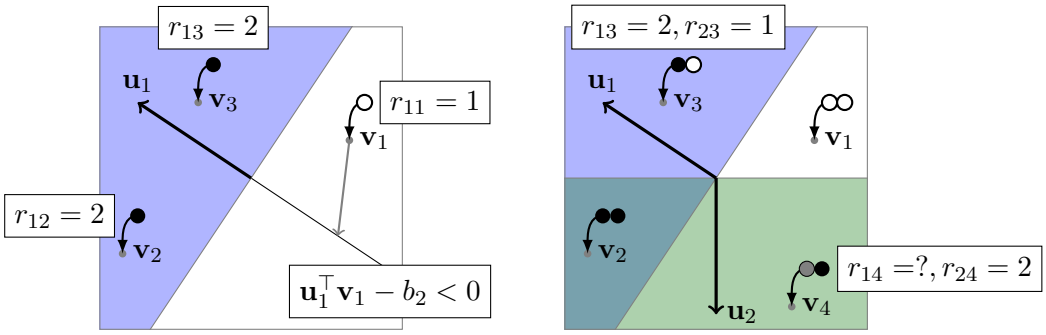

Figure 2: A visual interpretation of ordinal matrix factorization, in this case for binary matrices. Viewer factors $\mathbf{v}_n$ are interpreted as "data points" with multiple labels r_{mn} – possibly one for each item m – that are linearly classified with the item factors $\mathbf{u}_m$ as "weight vectors". Both $\mathbf{u}_m$ and $\mathbf{v}_n$ can be freely moved to give high probability to the data point labels. A dual interpretation switches the roles of $\mathbf{u}_m$ and $\mathbf{v}_n$.

It is also possible to give the hierarchical model an interpretation as a low rank approximation to the item-to-item (or equivalently user-to-user) based collaborative filtering. In item-to-item collaborative filtering, as for example employed by Amazon (Linden et al., 2003), the frequency of pairs of items that are bought by the same customer are recorded. Such a correlation matrix is also implicitly estimated in the present model, as can be seen by considering the joint prior distribution of a latent item vector for one user $\mathbf{h} = \mathbf{U}^\top \mathbf{v} + \boldsymbol{\epsilon}$ upon marginalizing the user factor $\mathbf{v}$ and $\boldsymbol{\epsilon}$:

$$p(\mathbf{h}|\mathbf{U}, \boldsymbol{\mu}_v, \boldsymbol{\Psi}_v, \gamma) = \mathcal{N}\left(\mathbf{h}\,;\, \mathbf{U}^\top \boldsymbol{\mu}_v,\, \mathbf{U}^\top \boldsymbol{\Psi}_v^{-1} \mathbf{U} + \mathbf{I}_M \gamma^{-1}\right) \,.$$

$\mathbf{U}^\top \boldsymbol{\Psi}_v^{-1} \mathbf{U} + \mathbf{I}_M \gamma^{-1}$ thus serves as a low rank approximation to the latent item-to-item covariance.

3. Gibbs sampling

Gibbs sampling is a MCMC method that sequentially samples from the conditional distributions of the model (Geman and Geman, 1984). Under some mild conditions – the Markov chain must be both aperiodic and irreducible – the samples produced will be from the posterior distribution (Robert and Casella, 2004). This section presents a Gibbs sampler in Algorithm 1, along with the conditional distributions that it requires. Apart from sampling for the latent variables h_{mn}, all the Gibbs updates are standard conjugate exponential updates.

Factors. The conditional distribution for each item factor $\mathbf{u}_m$ is Gaussian,

$$\mathbf{u}_m \sim \mathcal{N}\left(\mathbf{u}_m; \boldsymbol{\Sigma}_m \left[\boldsymbol{\Psi}_u \boldsymbol{\mu}_u + \gamma \sum_{n \in \Omega(m)} h_{mn} \mathbf{v}_n\right], \boldsymbol{\Sigma}_m\right)$$
$$\boldsymbol{\Sigma}_m = \left(\boldsymbol{\Psi}_u + \gamma \sum_{n \in \Omega(m)} \mathbf{v}_n \mathbf{v}_n^\top\right)^{-1},$$

where $\Omega(m)$ is the set of users that rated item m. Notice that the distribution for factor m only requires knowledge of γ and the variables directly connected with m: $\{(\mathbf{v}_n, h_{mn} | n \in \Omega(m)\}$. This suggests the order of the updates in Algorithm 1, where the latent variables are sampled for when needed and not stored in memory. A similar update, which relies on $\Pi(n)$, the set of items rated by user n, exists for user factor $\mathbf{v}_n$.

Latent variables. Sampling the conditional for the latent variable

$$p(h_{mn} | r_{mn}, \mathbf{u}_m, \mathbf{v}_n, \gamma) \propto p(r_{mn} | h_{mn})\, p(h_{mn} | \mathbf{u}_m, \mathbf{v}_n, \gamma)$$

requires one evaluation of a unit interval random number, one random Normal number, two evaluations of Φ and one of Φ^{-1}. When the likelihood function is a step function, the required Normal random number can be avoided. To derive the sampler we introduce the "noise-free" latent variable

$$\Phi(h - b) = \int \mathcal{N}(f; h, 1)\, \Theta(f - b)\, df$$

(see Albert and Chib, 1993). For any m and n, which is omitted here for brevity, the joint marginal distribution of r, f, and h, given $\mu = \mathbf{u}^\top \mathbf{v}$ and γ, is

$$p(r|f)\, p(f|h)\, p(h|\mu, \gamma) = \Big[\Theta(b_{r+1} - f) - \Theta(b_r - f)\Big]\, \mathcal{N}(f; h, 1)\, \mathcal{N}(h; \mu, \gamma^{-1}) \,. \tag{8}$$

The density $f, h | r, \mu, \gamma$ can be sampled from in two steps, $f | r, \mu, \gamma$ and $h | f, \mu, \gamma$. The distribution $f | r, \mu, \gamma$ is a truncated Normal

$$p(f | r, \mu, \gamma) = \frac{\mathcal{N}(f; \mu, 1 + \gamma^{-1}) \left[\Theta(b_{r+1} - f) - \Theta(b_r - f)\right]}{\Phi_{\max} - \Phi_{\min}}$$

with $\Phi_{\max} = \Phi\left(\frac{b_{r+1} - \mu}{\sqrt{1 + \gamma^{-1}}}\right)$ and $\Phi_{\min} = \Phi\left(\frac{b_r - \mu}{\sqrt{1 + \gamma^{-1}}}\right)$. A sample can be drawn from $p(f | r, \mu, \gamma)$ using the cumulative distribution

$$f = \mu + \sqrt{1 + \gamma^{-1}}\; \Phi^{-1}\Big(\Phi_{\min} + \text{rand}\,(\Phi_{\max} - \Phi_{\min})\Big) \,, \tag{9}$$

where "rand" gives a uniform random number between zero and its argument. Numerically, the Φ^{-1} step should be handled with care, as discussed in Appendix A. The desired sample is obtained from:

$$p(h | f, \mu, \gamma) = \mathcal{N}\left(h\,;\, (f + \gamma\mu)/(1 + \gamma),\, (1 + \gamma)^{-1}\right) \,.$$

Algorithm 1 Gibbs sampling

initialize $\mathbf{U}$, $\mathbf{V}$, $\boldsymbol{\mu}_u$, $\boldsymbol{\Psi}_u$, $\boldsymbol{\mu}_v$, $\boldsymbol{\Psi}_v$, γ
repeat
  for items $m = 1, \ldots, M$ in random order **do**
    sample $h_{mn}|r_{mn}, \mathbf{u}_m, \mathbf{v}_n, \gamma$ for each $n \in \Omega(m)$
    sample $\mathbf{u}_m|h_{mn}, \mathbf{v}_n, \gamma, \boldsymbol{\mu}_u, \boldsymbol{\Psi}_u$
  end for
  for users $n = 1, \ldots, N$ in random order **do**
    sample $h_{mn}|r_{mn}, \mathbf{u}_n, \mathbf{v}_m, \gamma$ for each $m \in \Pi(n)$
    sample $\mathbf{v}_n|h_{mn}, \mathbf{u}_m, \gamma, \boldsymbol{\mu}_v, \boldsymbol{\Psi}_v$
    collect statistics for γ: $\Delta_n = \sum_{m \in \Pi(n)} (h_{mn} - \mathbf{u}_m^\top \mathbf{v}_n)^2$
  end for
  sample $\boldsymbol{\mu}_u|\mathbf{U}, \boldsymbol{\Psi}_u$ and $\boldsymbol{\Psi}_u|\mathbf{U}$
  sample $\boldsymbol{\mu}_v|\mathbf{V}, \boldsymbol{\Psi}_v$ and $\boldsymbol{\Psi}_v|\mathbf{V}$
  sample $\gamma|\mathbf{H}, \mathbf{U}, \mathbf{V}$ using $b^{-1} = b_0^{-1} + \sum_n \Delta_n$
until sufficient samples have been taken

Normal-Wishart and Gamma. For the Normal-Wishart prior, $\boldsymbol{\mu}_u$ and $\boldsymbol{\Psi}_u$ are sampled from

$$\boldsymbol{\mu}_u, \boldsymbol{\Psi}_u \sim \mathcal{N}(\boldsymbol{\mu}_u; \boldsymbol{\mu}, (\beta \boldsymbol{\Psi}_u)^{-1}) \; \mathcal{W}(\boldsymbol{\Psi}_u; \mathbf{W}, \nu) \; ,$$

where

$$\begin{aligned}
\boldsymbol{\mu} &= \frac{\beta_0 \boldsymbol{\mu}_0 + M \overline{\mathbf{u}}}{\beta_0 + M} \\
\beta &= \beta_0 + M \\
\mathbf{W}^{-1} &= \mathbf{W}_0^{-1} + \frac{\beta_0 M}{\beta_0 + M} (\overline{\mathbf{u}} - \boldsymbol{\mu}_0)(\overline{\mathbf{u}} - \boldsymbol{\mu}_0)^\top + M\mathbf{S} \\
\nu &= \nu_0 + M \\
\mathbf{S} &= \frac{1}{M} \sum_m (\mathbf{u}_m - \overline{\mathbf{u}})(\mathbf{u}_m - \overline{\mathbf{u}})^\top \\
\overline{\mathbf{u}} &= \frac{1}{M} \sum_m \mathbf{u}_m \; .
\end{aligned}$$

Samples for $\boldsymbol{\mu}_v$ and $\boldsymbol{\Psi}_v$ can be drawn in the similar way. For the Gamma distribution γ is sampled from

$$\gamma \sim \Gamma(\gamma; a, b) \; , \qquad a = a_0 + \frac{|\mathcal{D}|}{2} \; , \qquad \frac{1}{b} = \frac{1}{b_0} + \frac{1}{2} \sum_{(m,n)} (h_{mn} - \mathbf{u}_m^\top \mathbf{v}_n)^2 \; .$$

Pseudo code. The pseudo code for the Gibbs sampler is given in Algorithm 1. The initial samples are usually discarded as a "burn-in" stage, and, together with the number of samples used, are discussed in Section 5. In practice, the algorithm's implementation should make use of an efficient data structure which stores the data twice, such that the

ratings in the sets $\Omega(m)$ and $\Pi(n)$ can easily be accessed. With the minor modification of replacing each occurrence of h_{mn} with r_{mn}, the sampler will sample from a model with a Gaussian likelihood instead of an ordinal regression one. For the Gaussian likelihood each of the for-loops can be completely parallelized. To parallelize the ordinal regression code, one needs to sample all the latent variables h_m, $(m, n) \in$ tr.set before each of the for-loops. This computation can also be performed in parallel. The algorithm may thus be made highly efficient on modern high memory GPUs and multi-core CPUs.

4. Variational Bayes

An alternative to Monte-Carlo estimates, such as those obtained in the previous section, is to use a parametric approximation. The posterior distribution, $p(\theta|\mathcal{D})$, is approximated by a simpler distribution, $q(\theta)$, which is used for further inference. A mean field approximation is often used, meaning that the approximating distribution is assumed to factorize. Variational Bayes (VB) is one popular algorithm for obtaining this approximation (Jordan et al., 1999). At each stage in the algorithm, one of the approximating factors is chosen, all other factors are fixed and the algorithm minimizes the Kullback-Leibler divergence $\mathrm{KL}(q(\theta)\|p(\theta|\mathcal{D}))$ by varying the given factor. The parameters used in obtaining this optimum will depend on statistics from the fixed factors.

In the case of the model used in this paper, a tractable mean field approximation is

$$q(\theta) = \prod_{(m,n)} q(h_{mn}) \prod_m q(\mathbf{u}_m) \prod_n q(\mathbf{v}_n)\; q(\boldsymbol{\mu}_u, \boldsymbol{\Psi}_u)\; q(\boldsymbol{\mu}_v, \boldsymbol{\Psi}_v)\; q(\gamma)$$

with the item and user factor variational distributions becoming Gaussian $q(\mathbf{u}_m) = \mathcal{N}(\mathbf{u}_m; \langle \mathbf{u}_m \rangle, \boldsymbol{\Sigma}_m)$ and $q(\mathbf{v}_n) = \mathcal{N}(\mathbf{v}_n; \langle \mathbf{v}_n \rangle, \boldsymbol{\Xi}_n)$. In this section angular brackets are used to denote an average with respect to the variational distribution. The factorized approximations for the hierarchical parameters follow the prior's Normal-Wishart form with

$$q(\boldsymbol{\mu}_u, \boldsymbol{\Psi}_u) = \mathcal{N}(\boldsymbol{\mu}_u; \boldsymbol{\mu}_{u_{\mathrm{VB}}}, (\beta_{u_{\mathrm{VB}}} \boldsymbol{\Psi}_u)^{-1})\, \mathcal{W}(\boldsymbol{\Psi}_u; \mathbf{W}_{u_{\mathrm{VB}}}, \nu_{u_{\mathrm{VB}}}) \;,$$

with a similar prior $q(\boldsymbol{\mu}_v, \boldsymbol{\Psi}_v)$, while $q(\gamma)$ is chosen as a Gamma density $q(\gamma) = \Gamma(\gamma;\, a_{\mathrm{VB}}, b_{\mathrm{VB}})$. As will be clear in the following, the variational distribution for h_{nm} is not needed explicitly because online computation of the first and second moments are sufficient for updating other variational distributions. The variational parameters to be kept track of are therefore $\langle \mathbf{u}_m \rangle$, $\langle \boldsymbol{\Sigma}_m \rangle$, $\langle \mathbf{v}_n \rangle$, and $\langle \boldsymbol{\Xi}_n \rangle$ for all m and n, and all parameters subscripted with "VB".

In the case of the mean field approximation, the algorithm for performing the iterative VB updates has the same structure as Algorithm 1, replacing all sampling steps with updates of sufficient statistics. Hence the pseudo-code is not repeated. One can show that the required updates are as follows:

Factors. The full update for each of the item factors is

$$
\begin{aligned}
q(\mathbf{u}_m) &= \mathcal{N}\left(\mathbf{u}_m;\, \langle \mathbf{u}_m \rangle,\, \boldsymbol{\Sigma}_m\right) \\
\langle \mathbf{u}_m \rangle &= \boldsymbol{\Sigma}_m \left[\langle \boldsymbol{\Psi}_u \boldsymbol{\mu}_u \rangle + \langle \gamma \rangle \sum_{n \in \Omega(m)} \langle h_{nm} \rangle \langle \mathbf{v}_n \rangle \right] \\
\boldsymbol{\Sigma}_m &= \left(\langle \boldsymbol{\Psi}_u \rangle + \langle \gamma \rangle \sum_{n \in \Omega(m)} \left(\boldsymbol{\Xi}_n + \langle \mathbf{v}_n \rangle \langle \mathbf{v}_n^\top \rangle \right) \right)^{-1},
\end{aligned}
$$

with $\langle \boldsymbol{\Psi}_u \boldsymbol{\mu}_u \rangle = \langle \boldsymbol{\Psi}_u \rangle \langle \boldsymbol{\mu}_u \rangle = \nu_{u_{\mathrm{VB}}} \mathbf{W}_{u_{\mathrm{VB}}} \boldsymbol{\mu}_{u_{\mathrm{VB}}}$. As is true for the Gibbs sampler, similar updates to the one above exist for the user factors, $\mathbf{v}_n$.

When either M or N is large, memory constraints may not allow an entire covariance matrix to be stored for each $q(\mathbf{u}_m)$ and $q(\mathbf{v}_n)$. In this case $\boldsymbol{\Sigma}_n$ and $\boldsymbol{\Xi}_n$ can be constrained to be diagonal matrices such that $q(\mathbf{u}_m) = \prod_{i=1}^{K} q(u_{mi})$, for example, factorizes completely. Under a Normal-Wishart prior the updates will take a slightly different form. With $q(u_{mi}) = \mathcal{N}(u_{mi}; \langle u_{mi} \rangle, \Sigma_{mi})$ they are

$$
\begin{aligned}
\Sigma_{mi} &= \left(\langle \boldsymbol{\Psi}_u \rangle_{ii} + \langle \gamma \rangle \sum_{n \in \Omega(m)} \left(\Xi_{ni} + \langle v_{ni} \rangle^2 \right) \right)^{-1} \\
\langle u_{mi} \rangle &= \Sigma_{mi} \Bigg\{ \langle \boldsymbol{\Psi}_u \boldsymbol{\mu}_u \rangle_i + \langle \gamma \rangle \sum_{n \in \Omega(m)} \langle v_{ni} \rangle \Big(\langle h_{nm} \rangle + \cdots \\
&\qquad \langle u_{mi} \rangle \langle v_{ni} \rangle - \langle \mathbf{u}_m^\top \rangle \langle \mathbf{v}_n \rangle \Big) + \langle u_{mi} \rangle \langle \boldsymbol{\Psi}_u \rangle_{ii} - \langle \mathbf{u}_m^\top \rangle \langle \boldsymbol{\Psi}_u \rangle_i \Bigg\}.
\end{aligned}
$$

In the above update $\langle \boldsymbol{\Psi}_u \rangle_i$ is used to indicate column i of $\langle \boldsymbol{\Psi}_u \rangle$, and $\langle \boldsymbol{\Psi}_u \rangle_{ii}$ diagonal element (i, i). The notation $\langle u_{mi} \rangle \langle v_{ni} \rangle - \langle \mathbf{u}_m^\top \rangle \langle \mathbf{v}_n \rangle$ is equivalent to $-\sum_{j \neq i} \langle u_{mj} \rangle \langle v_{nj} \rangle$, while the notation $\langle u_{mi} \rangle \langle \boldsymbol{\Psi}_u \rangle_{ii} - \langle \mathbf{u}_m^\top \rangle \langle \boldsymbol{\Psi}_u \rangle_i$ expands to a similar expression.

Latent variables. The mean and variance of h_{mn} are determined when needed in other updates from

$$
\begin{aligned}
q(h_{mn}) &\propto p(r_{mn} | h_{mn}) \cdots \\
&\qquad \exp\left(\Big\langle \log p(h_{mn} | \mathbf{u}_m, \mathbf{v}_n, \gamma) \Big\rangle_{q(\mathbf{u}_m)\, q(\mathbf{v}_n)\, q(\gamma)} \right) \\
&\propto p(r_{mn} | h_{mn})\, \mathcal{N}(h_{mn}; \langle \mathbf{u}_m^\top \rangle \langle \mathbf{v}_n \rangle,\, \langle \gamma \rangle^{-1}).
\end{aligned}
$$

Define $\mu = \langle \mathbf{u}_m^\top \rangle \langle \mathbf{v}_n \rangle$, and let γ be short for $\langle \gamma \rangle$, and $\mathcal{N}(z)$ short for $\mathcal{N}(z; 0, 1)$. If b_r and b_{r+1} are the boundaries associated with r_{mn}, and $z_r = (\mu - b_r)/\sqrt{1 + \gamma^{-1}}$, the explicit expressions for $\langle h_{nm} \rangle$ and $\langle h_{nm}^2 \rangle$ are

$$
\begin{aligned}
\langle h_{mn} \rangle &= \mu + \frac{\gamma^{-1}}{\sqrt{1 + \gamma^{-1}}} \frac{\mathcal{N}(z_r) - \mathcal{N}(z_{r+1})}{\Phi(z_r) - \Phi(z_{r+1})} \\
\langle h_{mn}^2 \rangle &= 2\mu \langle h_{mn} \rangle - \mu^2 + \gamma^{-1} \cdots \\
&\qquad - \frac{\gamma^{-2}}{1 + \gamma^{-1}} \frac{z_r\, \mathcal{N}(z_r) - z_{r+1}\, \mathcal{N}(z_{r+1})}{\Phi(z_r) - \Phi(z_{r+1})}.
\end{aligned}
$$

Appendix A gives the asymptotic forms that should be used when z_r or z_{r+1} is sufficiently small or large.

Normal-Wishart and Gamma. The variational parameters for $q(\boldsymbol{\mu}_u, \boldsymbol{\Psi}_u)$ are updated with

$$\begin{aligned}
\boldsymbol{\mu}_{u\text{VB}} &= \frac{\beta_0 \boldsymbol{\mu}_0 + M\overline{\langle \mathbf{u} \rangle}}{\beta_0 + M} \\
\beta_{u\text{VB}} &= \beta_0 + M \\
\mathbf{W}_{u\text{VB}}^{-1} &= \mathbf{W}_0^{-1} + \frac{\beta_0 M}{\beta_0 + M} \left(\overline{\langle \mathbf{u} \rangle} - \boldsymbol{\mu}_0 \right) \left(\overline{\langle \mathbf{u} \rangle} - \boldsymbol{\mu}_0 \right)^\top + M\mathbf{S} \\
\nu_{u\text{VB}} &= \nu_0 + M \\
\mathbf{S} &= \frac{1}{M} \sum_{m=1}^{M} \left\{ \boldsymbol{\Sigma}_m + \left(\langle \mathbf{u}_m \rangle - \overline{\langle \mathbf{u} \rangle} \right) \left(\langle \mathbf{u}_m \rangle - \overline{\langle \mathbf{u} \rangle} \right)^\top \right\} \\
\overline{\langle \mathbf{u} \rangle} &= \frac{1}{M} \sum_{m=1}^{M} \langle \mathbf{u}_m \rangle \ ,
\end{aligned}$$

while the parameters of $q(\boldsymbol{\mu}_v, \boldsymbol{\Psi}_v)$ follow a similar update. The updates for the parameters of $q(\gamma)$ are

$$\begin{aligned}
a_{\text{VB}} &= a_0 + \frac{|\mathcal{D}|}{2} \\
b_{\text{VB}} &= \frac{1}{b_0} + \frac{1}{2} \sum_{(m,n)} \left[\langle h_{mn}^2 \rangle - 2 \langle h_{mn} \rangle \langle \mathbf{u}_m^\top \rangle \langle \mathbf{v}_n \rangle + \left\langle \mathbf{u}_m^\top \mathbf{v}_n \mathbf{v}_n^\top \mathbf{u}_m \right\rangle \right] \\
&= \frac{1}{b_0} + \frac{1}{2} \sum_{(m,n)} \Bigg[\langle h_{mn}^2 \rangle - \langle h_{mn} \rangle^2 + \left(\langle h_{mn} \rangle - \langle \mathbf{u}_m^\top \rangle \langle \mathbf{v}_n \rangle \right)^2 \cdots \\
&\qquad + \operatorname{tr}[\boldsymbol{\Sigma}_m \boldsymbol{\Xi}_n] + \langle \mathbf{u}_m^\top \rangle \boldsymbol{\Xi}_n \langle \mathbf{u}_m \rangle + \langle \mathbf{v}_n^\top \rangle \boldsymbol{\Sigma}_m \langle \mathbf{v}_n \rangle \Bigg] \ .
\end{aligned}$$

When $\boldsymbol{\Sigma}_m$ and $\boldsymbol{\Xi}_n$ are diagonal matrices, the update scales linearly with K.

5. Evaluation

The proposed model is evaluated by testing its predictive performance on the Netflix data set. The data set contains around *100 million* ratings from $N = 480,189$ users on $M = 17,770$ movie titles. Each rating is a number of stars 1 to 5, and is used as the ranked observation at the relevant point in $\mathbf{R}$. The amount of data for particular users and movies is varied, as some users rated fewer than ten movies, while others took the time to rate up to a few thousand movies. A test or "qualifying" set of almost three million user–movie pairs for which the ratings are withheld. These (withheld) ratings are taken from the most recent ratings in the period over which the data was collected. Algorithms are benchmarked by their RMSE computed over an unknown half of the test set.

	γ Gibbs				γ VB	
K	0.08	0.09	0.1	inferred	0.1	inferred
50	0.8959	0.8958	0.8961	0.8970	0.8986	0.8983
100	0.8935	0.8930	0.8928	0.8943	0.8987	0.8989
200	0.8927	0.8917	0.8913	0.8934	0.9005	0.9021

Table 1: RMSE on the Netflix qualifying set for the hierarchical model, using Gibbs sampling and VB for inference. The same results are plotted in Figure 3.

5.1 Hyperparameter settings for the Netflix data.

The first hyperparameters which must be chosen are the boundaries for the different ranks. In this work the boundaries were set to be equidistant with boundaries $(b_1, \ldots, b_6) = (-\infty, -6, -2, 2, 6, \infty)$, although any other equidistant choice would lead to the same result after an appropriate re-scaling of the remaining parameters. Although learning the scaling between the b's would reflect the relative frequency of particular ratings appearing in the data, it is not included in this work.

The precise choice of the hyperparameters of the Normal-Wishart prior for the mean and variance of the factors, $\mathcal{N}(\boldsymbol{\mu}_u; \boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Psi}_u)^{-1})\mathcal{W}(\boldsymbol{\Psi}_u; \mathbf{W}_0, \nu_0)$, is not crucial as the copious amounts of data for learning will override any reasonably weak prior. To reflect this, the mean of the mean prior is set to $\boldsymbol{\mu}_0 = \mathbf{0}$, the mean-precision coupling parameter is set to $\beta_0 = 1$, the precision matrix is set to the identity matrix $\mathbf{W}_0 = \mathbf{I}$, and the degrees of freedom reflect the dimensionality $\nu_0 = K + 1$.

The most critical parameter in the problem is γ, the inverse variance of the h latent variable distribution. This parameter largely determines the degree of fitting to the training data. As an initial choice $a_0 = 10$ and $b_0 = 10^{-2}$ were used in the Gamma distribution prior, corresponding to the prior $p(h_{mn}|\mathbf{u}_m, \mathbf{v}_n, \gamma)$ in (3) having an expected variance of $\langle \gamma^{-1} \rangle = 1/[(a_0 - 1)b_0] = 9$. The motivation for this choice is that it roughly gives a baseline expected RMSE of $\sqrt{\langle \gamma^{-1} \rangle}/\delta b = 3/4$, which is substantially below the Netflix prize goal. When these parameter settings were used in Gibbs sampling a $K = 50$ model, samples for γ converged to around 0.119 with very small fluctuations, since it is well-determined by the large amount of data. It transpired that keeping γ fixed at a value $\sim 0.08 - 0.1$ somewhat improves the test error performance. This can be ascribed to the fact that the training and qualifying data are not sampled from entirely the same distribution. The qualifying set contains more recent ratings and there is a substantial upward drift in the ratings over time.

The inclusion of the hierarchical prior in the model plays a role in the model's predictive accuracy. Gibbs sampling with $\gamma = 0.1$ and a hierarchical prior ($K = 50$) gives a RMSE of 0.8961 on the Netflix quiz set. Removing the hyper-priors by additionally keeping $\boldsymbol{\Psi}_u$, $\boldsymbol{\mu}_u$, $\boldsymbol{\Psi}_v$, and $\boldsymbol{\mu}_v$ *fixed* at their mean values gives a RMSE of 0.9091.

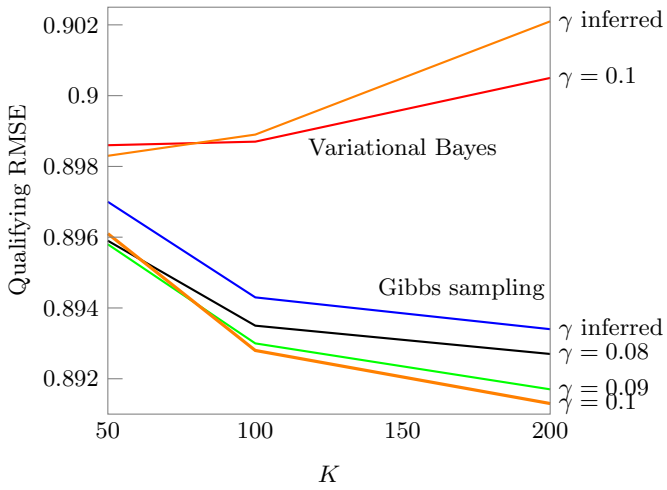

Figure 3: RMSE on the qualifying set as a function of K for different γ-settings. The four lower lines are for Gibbs sampling and the two upper lines for VB. The same results are given in Table 1.

5.2 Performance results

The hidden ratings in the Netflix test set were inferred using Gibbs sampling and VB for a number latent dimensions, $K = 50, 100, 200$, and γ settings $\gamma = 0.08$, 0.09, 0.1, and one setting in which γ is also inferred. The results are summarized in Table 1 and Figure 3. When the latent factor dimensionality K is increased, a higher precision γ gives better performance. The Gibbs-sampled results provide further evidence that proper regularization in models with more parameters than the data set size $|\mathcal{D}|$ is possible with Bayesian averaging (Neal, 1996).

Effect of likelihood and support. When the results are compared to the Gaussian-likelihood factor model of (Salakhutdinov and Mnih, 2008a), the ordinal likelihood gives some improvement (compare for example the 0.8958, 0.8928, and 0.8913 RMSE for $K = 50, 100$ and 200 to their Gaussian likelihood's $0.8989, 0.8965$ and 0.8954 RMSE for $K = 60, 150$ and 300).

However, one of the advantages of using an ordinal likelihood instead of a Gaussian is that the model provides an estimate of the probability for each possible rating. This advantage is not evaluated by the RMSE metric, but does become evident when using alternative metrics such as the mean absolute error (MAE), the empirical mean of $|\hat{r} - r_{\text{true}}|$. Figure 4 plots the two metrics when using both the ordinal and Gaussian likelihood. The results are grouped according to the movie support in order to analyse the effect of the number of movies on predictions.

As expected, the figure shows that the ordinal likelihood results in a larger improvement when evaluated using the MAE. The largest improvement is obtained on movies with few ratings, presumably because there are less averaging effects to help the Gaussian model.

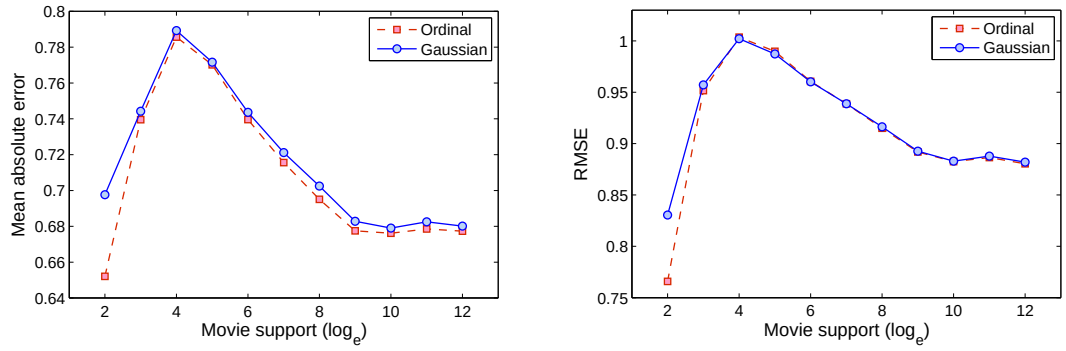

Figure 4: RMSE and MAE as a function of the movie support using ordinal and Gaussian likelihoods. The movies are *grouped by* $\lfloor \log |\Omega(m)| \rfloor$. The estimates are obtained on the Netflix qualifying set, using Gibbs-sampled $K = 50$ models.

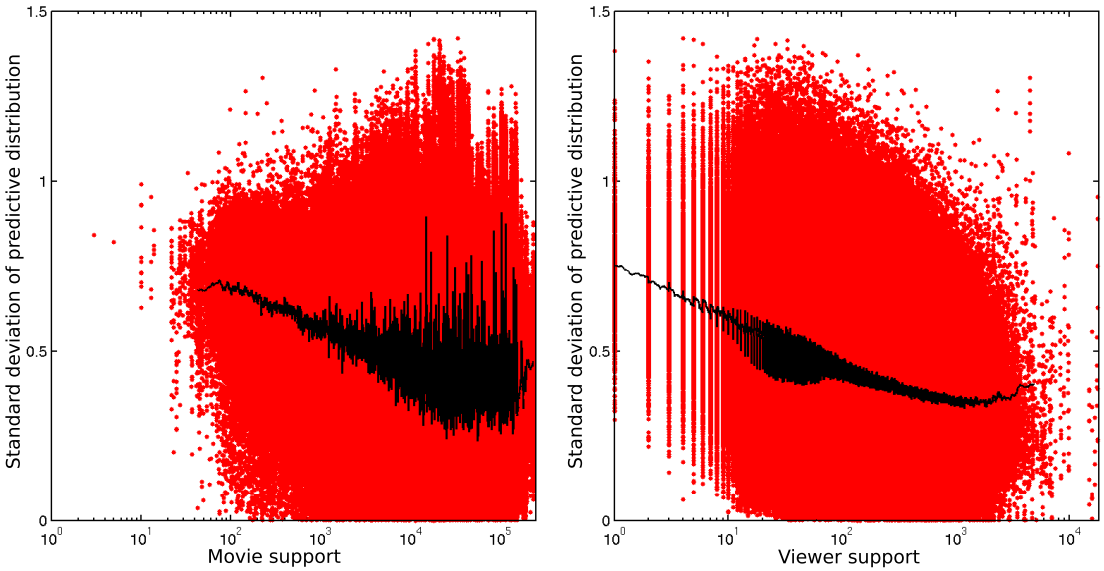

Figure 5: Predictive uncertainty $\sqrt{\langle r_{mn}^2 \rangle - \langle r_{mn} \rangle^2}$ of the qualifying set points (dots) as a function of the movie- (left) and viewer-support (right). The dark lines show a windowed average (window size 1000).

Interestingly, both models achieve best results on movies with low support. One explanation for this may be that the movies are only being watched by a handful of connoisseurs who tend to rate them similarly. As the support increases the metrics both deteriorate at first and then improve as the extra data improves the estimates.

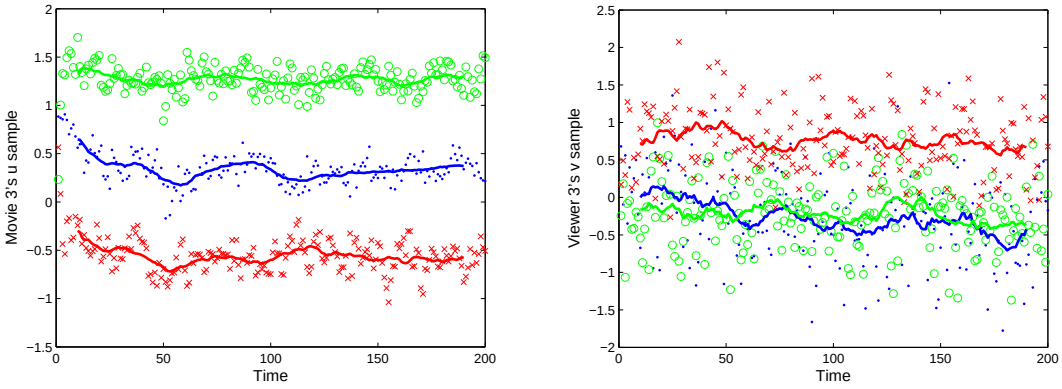

Figure 6: The samples for six of the five million model parameters required for a "small" model with $K = 10$. The samples for the first three components of $\mathbf{u}_3$ for movie 3, i.e. u_{13}, u_{23}, and u_{33}, are shown at the top. Movie 3 had $\Omega(3) = 2011$ ratings. The samples for the first three components of $\mathbf{v}_3$ are shown at the bottom.Viewer 3 rated $\Pi(3) = 97$ movies. The overlaid lines indicate a windowed average over 20 samples.

Predictive variance. The effect of averaging over the posterior can be visualized by plotting the predictive uncertainty $\sqrt{\langle r_{mn}^2 \rangle - \langle r_{mn} \rangle^2}$ for *each* withheld movie–user entry (m, n) in the qualifying set as a function of the "support". The "movie support" is defined as the number of viewers $|\Omega(m)|$ for a particular movie m in question. Figure 5 shows the movie- (top) and viewer support (bottom). There is a clear trend that the uncertainty decreases with the support, and, barring movies m with $|\Omega(m)| \lesssim 50$, correlates with the empirical errors in Figure 4.

Gibbs sampling and burn-in. The Gibbs sampler converges fast in terms of prediction accuracy. (As an example, the samples for the first three components of a user and movie's latent vectors with $K = 10$ are illustrated in Figure 6.) This paper used a burn-in of 20 updates of all parameters starting from random initial values, and collected posterior statistics for another 180 updates. An additional 180 steps lead to a performance increase of around 0.0005 RMSE.

The comparison between Gibbs sampling and variational Bayes on such a large task gives rise to several conclusions that are applicable to other similar settings (factor models with relative high noise levels). VB and Gibbs have the same computational complexity per update as identical matrix operations dominate the update of variational and conditional distributions. Variational inference for the smaller factor sizes ($K = 50$) showed promising results, but unfortunately overfitted for larger models. VB might, like penalized likelihood models, be able to achieve better performance with careful parameter tuning, but as this is time consuming for large problems Gibbs sampling can generally be recommended.

Alternative and similar approaches. There exists a large number of collaborative filtering approaches in the machine learning literature that are all evaluated on the Netflix data set. They vary greatly in nature and purpose: some focus on runtime complexity; some aim to be general, while other solutions are tailored to peculiarities of the Netflix data. Table 2 provides an overview of the RMSE scores available for some models.[2] Even though results of matrix factorization methods are not reported for the same K-setting, a comparison of the RMSE scores provides an overview of the diversity of available techniques.

There are approaches that strongly resemble this work. In Table 2, the Matchbox system of Stern et al. (2009) also employs an ordinal likelihood function, and can additionally regress against user and item features. Most other systems require a Gaussian likelihood: Porteous et al. (2010)'s BMFSI also regresses against user and item features, with the addition of Dirichlet process mixture priors on $\mathbf{u}_m$ and $\mathbf{v}_n$. Another mixture is introduced in MMMF (Mackey et al., 2010), where a user–item bias is added to $\mathbf{u}_m^\top \mathbf{v}_n$. The bias is obtained by combining a discrete "user cluster" and "item cluster" assignment; this captures one of a fixed number of possible predispositions towards liking or disliking each item, irrespective of the static latent factor parameterization. Salakhutdinov and Mnih (2008a) Gibbs-sample a hierarchical model with a Gaussian likelihood, while Lim and Teh (2007) make a variational approximation to the parameters' posterior distribution. The addition of a local neighbour-based correction term to predictions obtained from global matrix factorization improves predictions on the Netflix test set; this is illustrated by Takács et al. (2009).

The best results in Table 2 are given by non-parametric extensions, where Zhu et al. (2009) and Yu et al. (2009a) present a flexible generalization of low-rank matrix factorization to an infinite relational function. Not only are correlations between the components (factors) of $\mathbf{u}_m$ explicitly learned, but also correlations between all items.

6. Conclusion and outlook

This paper has proposed a hierarchical model for ordinal matrix factorization. A minimal model was used to keep the message as clean as possible, although the results could plausibly be improved by blending with other models, as was shown by (Bell et al., 2007), and by taking temporal effects into account (Koren, 2009, Töscher et al., 2009). Some ideas that we are currently investigating are to include biases in the model $h_{mn} = \mathbf{u}_m^\top \mathbf{v}_n + a_m + b_n + \epsilon_{mn}$; to adapt the parameters of the ordinal likelihood function (there is no reason to *a priori* believe that an equidistant setting for $\mathbf{b}$ is the best); to generalize the Normal-Wishart priors to a mixture of Normal-Wisharts to capture more subtle factor dependencies (Porteous et al., 2010).

One important shortcoming of the model is made particularly evident by Figure 3. In the figure, the pre-specified settings of the noise parameter are shown to outperform the inferred γ. This may be because the inferred γ will be highly affected by the most popular movies, where the distribution of ratings may well be more concentrated. The movies with fewer ratings may be exactly the type which polarize opinion, leading to a higher variance. An important extension would therefore be to let the the noise parameter be movie dependent $\gamma \to \gamma_m$.

2. Only "single model" results are reported here. The best Netflix results are achieved through a weighted combination of many diverse models.

Method	Reference	RMSE
Cinematch	Netflix	0.9514
PMF	Salakhutdinov and Mnih (2008b)	0.9170
PMF-VB	Lim and Teh (2007)	0.9141
Matchbox	Stern et al. (2009)	0.914
RBM	Salakhutdinov et al. (2007)	∼0.907
BPMF	Salakhutdinov and Mnih (2008a)	0.8954
MMMF	Mackey et al. (2010)	0.8929
NPCA	Yu et al. (2009b)	0.8926
OMF	**This work**	**0.8913**
BRISMF	Takács et al. (2009)	0.8904
BMFSI	Porteous et al. (2010)	0.8875
BSRM	Zhu et al. (2009)	0.8874
NREM	Yu et al. (2009a)	0.8853

Table 2: RMSE on the Netflix qualifying set for a variety of models: Probabilistic Matrix Factorization (PMF); PMF with Variational Bayes (PMF-VB); Matchbox (ordinal likelihood PMF with expectation propagation and VB factor graph messages); Restricted Boltzmann machines (RBM); Bayesian PMF (BPMF); Mixed membership matrix factorization (MMMF); Nonparametric principal component analysis (NPCA); Ordinal matrix factorization (OMF); Biased regularized incremental simultaneous matrix factorization (BRISMF); Bayesian matrix factorization with side information (BMFSI); Bayesian stochastic relational model (BSRM); Nonparametric random effects model (NREM).

Such modifications can be implemented straight forwardly in Gibbs sampling and should not substantially change convergence times. A final extension is to incorporate additional information from the test set, which doesn't provide additional ratings, but reveals the identities of some other movies seen by certain users.

Appendix A. Asymptotics of the error function

Some care is needed to obtain numerically stable samples from, and evaluations of, probit-based likelihoods.

Gibbs sampling. When sampling $p(f|r,\mu,\gamma)$, as was done in (9), a sample is drawn from a standard Gaussian $\mathcal{N}(z)$ that is truncated between $z_{\min} = -z_r = (b_r - \mu)/\sqrt{1+\gamma^{-1}}$ and $z_{\max} = -z_{r+1}$, with

$$z = \Phi^{-1}\Big(\Phi_{\min} + \mathrm{rand}\,(\Phi_{\max} - \Phi_{\min})\Big) \ .$$

This sample is then scaled with a standard deviation and mean-shifted in (9) to obtain f. A numerical error arises in this procedure when the argument of Φ^{-1} evaluates to zero to machine precision. This happens when $z_{\min}$ and $z_{\max}$ are both sufficiently *small* or *large* so that $\Phi_{\min} = \Phi(z_{\min})$ and $\Phi_{\max} = \Phi(z_{\max})$ are either both equal to (or close to) zero or one with finite machine precision. In these cases the distribution of z is truncated in the tail of the Gaussian, and will be strongly peaked at one of the ends of the interval $z \in (z_{\min}, z_{\max})$. We may therefore deterministically set $z = z_{\max}$ if $z_{\max} \leq -5$ or $z = z_{\min}$ if $z_{\min} \geq 5$, where 5 (standard deviations) is large enough for a double precision Φ^{-1} evaluation to be precise.

Variational Bayes. The statistic

$$\langle h \rangle = \mu + \frac{\gamma^{-1}}{\sqrt{1+\gamma^{-1}}} \frac{\mathcal{N}(z_{\max}) - \mathcal{N}(z_{\min})}{\Phi(z_{\max}) - \Phi(z_{\min})} \tag{10}$$

becomes numerically unstable when $z_{\min}$ is sufficiently large or $z_{\max}$ is sufficiently small, and asymptotic expansions are needed for the last term for the two limits $z_{\max} \to -\infty$ and $z_{\min} \to \infty$. As Φ is a monotonically increasing function and

$$\begin{aligned}\Phi(z_{\max}) - \Phi(z_{\min}) &= (1 - \Phi(-z_{\max})) - (1 - \Phi(-z_{\min})) \\ &= \Phi(-z_{\min}) - \Phi(-z_{\max}) \ ,\end{aligned}$$

the largest term in numerator (by an exponential factor) will be $\Phi(z_{\max})$ for $z_{\max} \to -\infty$ and $\Phi(-z_{\min})$ for $z_{\min} \to \infty$. Using a similar argument, one of the terms in the numerator can be dropped.

The application of l'Hôpital's rule leads to $\mathcal{N}(z)/\Phi(z) \to -z$ as $z \to \pm\infty$, but a more precise approximation can be obtained by keeping additional terms of the power series of $\Phi(z)$ for $z \to -\infty$,

$$\Phi(z) = -\mathcal{N}(z)\left(\frac{1}{z} - \frac{1}{z^3} + \frac{3}{z^5} - \frac{15}{z^7} + \cdots\right) \ .$$

When the power series is combined with the geometric series $\frac{-z}{1-a} = -z(1 + a + a^2 + \cdots)$, where $a = \frac{1}{z^2} - \frac{3}{z^4} + \frac{15}{z^6} - \cdots$, and the dominant terms kept, the approximation becomes

$$\frac{\mathcal{N}(z)}{\Phi(z)} \to -z - \frac{1}{z} + \frac{2}{z^3} \quad \text{for} \quad z \to -\infty \ .$$

We therefore get $\langle h \rangle = \mu - \gamma^{-1}/\sqrt{1+\gamma^{-1}}\,(z_{\max} + 1/z_{\max} - 2/z_{\max}^3)$ for $z_{\max} \to -\infty$ and $\langle h \rangle = \mu - \gamma^{-1}/\sqrt{1+\gamma^{-1}}\,(z_{\min} + 1/z_{\min} - 2/z_{\min}^3)$ for $z_{\min} \to \infty$. To double precision one may use quite large values for swapping to the asymptotic regime, for example $z_{\max} < -35$ and $z_{\min} > 35$.

References

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

A. Ansari, S. Essegaier, and R. Kohli. Internet recommendation systems. *Journal of Marketing Research*, pages 363–375, 2000.

R. M. Bell and Y. Koren. Improved neighborhood-based collaborative filtering. In *Proceedings of KDD Cup and Workshop*. 2007.

R. M. Bell, Y. Koren, and C. Volinsky. The BellKor solution to the Netflix prize. Technical report, AT&T Labs–Research, 2007.

M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(1):155–173, 2007.

W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

T. Hofmann. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, pages 289–296, 1999.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

Y. Koren. The BellKor solution to the Netflix Grand Prize. Technical report, 2009.

N. D. Lawrence and R. Urtasun. Non-linear matrix factorization with Gaussian processes. In L. Bottou and M. Littman, editors, *Proceedings of the International Conference in Machine Learning*. Morgan Kauffman, San Francisco, CA, 2009.

Y. J. Lim and Y. W. Teh. Variational Bayesian approach to movie rating prediction. In *Proceedings of KDD Cup and Workshop*. 2007.

G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003. ISSN 1089-7801.

L. Mackey, D. Weiss, and M. I. Jordan. Mixed membership matrix factorization. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning*, pages 711–718, 2010.

B. Marlin. Modeling user rating profiles for collaborative filtering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.

P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, and H. Mannila. The discrete basis problem. *IEEE Transactions on Knowledge and Data Engineering*, (10):1348–1362, 2008.

R. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, New York, 1996.

M. Piotte and M. Chabbert. The Pragmatic Theory solution to the Netflix Grand Prize. Technical report, 2009.

I. Porteous, A. Asuncion, and M. Welling. Bayesian matrix factorization with side information and Dirichlet process mixtures. In *AAAI Conference on Artificial Intelligence*, 2010.

J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 713–719, 2005.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, second edition, 2004.

R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th International Conference on Machine Learning*, pages 880–887, 2008a.

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

R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the International Conference on Machine Learning*, volume 24, pages 791–798, 2007.

B. H. Shen, S. Ji, and J. Ye. Mining discrete patterns via binary matrix factorization. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 757–766, 2009.

N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. *Advances in Neural Information Processing Systems*, 17:1329–1336, 2005.

D. H. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online Bayesian recommendations. In *WWW*, pages 111–120, 2009.

S. S. Stevens. On the theory of scales of measurement. *Science*, 103(2684):677–680, 1946.

G. Takács, I. Pilászy, B. Németh, and D. Tikk. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656, 2009.

A. Töscher, M. Jahrer, and R. Bell. The BigChaos solution to the Netflix Grand Prize. Technical report, 2009.

K. Yu, J. Lafferty, S. Zhu, and Y. Gong. Large-scale collaborative prediction using a nonparametric random effects model. In L. Bottou and M. Littman, editors, *Proceedings of the International Conference in Machine Learning*. Morgan Kauffman, San Francisco, CA, 2009a.

K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast nonparametric matrix factorization for large-scale collaborative filtering. In *Proceedings of the 32nd international ACM SIGIR conference*, pages 211–218, 2009b.

Z. Y. Zhang, T. Li, C. Ding, X. W. Ren, and X. S. Zhang. Binary matrix factorization for analyzing gene expression data. *Data Mining and Knowledge Discovery*, pages 1–25, 2009.

S. Zhu, K. Yu, and Y. Gong. Stochastic relational models for large-scale dyadic data using MCMC. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1993–2000. 2009.