Journal of Machine Learning Research 1 (2005) 1-28 Submitted xxxx; Revised xxxx; Published xx/xx

Ulrich Paquet ULRICH@CANTAB.NET
Blaise Thomson BRMT2@CAM.AC.UK
Computer Laboratory
University of Cambridge
Cambridge CB3 0FD, United Kingdom

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

Editor: xxxx

Abstract

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

1. Introduction

- Discuss the basic collaborative filtering task.
- Discuss low rank what has been especially shortcomings Gaussian likelihood and MAP. Of course mention honorable exceptions.
- Discuss alternatives to low rank and discuss state-of-the-art on netflix.
- I would like this paper to focus entirely on our model and not making so many comparisons. We limit to citing other especially icml Gibbs paper.

Training the model generally involved finding the best rank K approximation to the $N \times M$ target matrix R. A number of probabilistic factor-based models have been proposed.

1. Hofmann (1999) does Probabilistic Latent Semantic Analysis.

Other collaborative filtering work:

1. Marlin (2004) does:

The remainder of the paper is organized as follows: Section 2 describes the Bayesian hierarchical model, Sections 3 and 4 describe inference in this model with Gibbs sampling and variational Bayes, respectively. Section 5 describes the experiments with the model on the Netflix prize task. We conclude in Section 6.

2. Probabilistic model

There are three elements in the hierarchical Bayesian model we use for the collaborative filtering: 1. An ordinal regression likelihood function which maps a latent variable h to probabilities for the discrete ranks r in such a way that the ordering is preserved, 2. a probabilistic model for latent variable in terms of a low rank matrix factorization and 3. hierarchical prior distributions for the low rank matrices. These elements are described in the following.

2.1 Ordinal regression likelihood

Ordinal regression arises when a choice of preference is made. An item with a five-star rating is generally regarded to be 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 preferences; we can merely say that class A is preferred to class B, which is preferred to C, etc. 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 rating can be computed.

For ranks or ratings $r = 1, \ldots, R$, partition the real line 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. If some space **may use another word: 'space' is a mathematically loaded one!**—in this paper entries in a low-rank matrix—is mapped to values f on this line with some continuous mapping, the *stochastic ordering on the space is preserved*. 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

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

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. Another interpretation of this likelihood function is to view $p(r \geq r'|h) = \Phi(h - b_r)$, so that

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

When only two ranks are present, (1) becomes the familiar binary classification likelihood[1].

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.

Let r_{mn} denote the rank for item (movie) $m = 1, \ldots, M$ and user (viewer) $n = 1, \ldots, N$. The ordinal regression model maps a continuous latent variable h_{mn} to probabilities $p(r_{nm}|h_{mn})$. **For generality we may include the set $\mathbf{b}_m$ for each item m to the model parameters. We keep them fixed so skip for now.** The probability of observing ranking r_{mn} can also be written as

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

using the δ-function to pick out the $r = r_{mn}$ term. The likelihood defined over a training set of ranks $\mathcal{D} = \{r_{mn}|(m,n) \in \text{tr.set}\}$ is

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

2.2 Low rank matrix factorization

A simple popular choice **Refs here** for modelling the latent variable is matrix factorization

$$h_{mn} = \mathbf{u}_m \cdot \mathbf{v}_n + \epsilon_{mn}$$

with ϵ_{mn} being the residual and $\mathbf{u}_m$ and $\mathbf{v}_n$ being the vector (factor of length K) associated with items m and user n, respectively. In matrix form $\mathbf{H} = \mathbf{U}^T\mathbf{V} + \boldsymbol{\epsilon}$, with $\mathbf{U}$ being a $K \times N$ user coefficient matrix, and $\mathbf{V}$ a $K \times M$ item coefficient matrix.

The model can be justified by the fact that the decomposition is low-rank, i.e. K is much smaller than both M and N. This means that we performing lossy data compression and thereby hopefully extracting the essential structure in the data. In simple low rank decomposition approaches, prediction on a test case (m', n') is performed by simple vector multiplication $\mathbf{u}_{m'} \cdot \mathbf{v}_{n'}$. In the Bayesian approach adopted here the predictive distribution still relies on such a dot product.

Recently it has been observed by Salakhutdinov and Mnih (2008) that with a proper regularization through Bayesian averaging it is advantageous to use a low rank decomposition with more parameters $K * (M + N)$ than ratings $|\mathcal{D}|$, the data set size or number of non-missing entries in $\mathbf{H}$.

For the latent variables we use a simple normal distribution with a shared variance γ^{-1} for all data points:

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

It turns out the performance of the model will essentially only depend upon the choice of the γ parameter and K, see Section 5.

2.3 Hierarchical Bayesian priors

The model parameters we need to specify priors for are $\mathbf{U}$, $\mathbf{V}$ and γ. We will use a standard Bayesian hierarchical treatment which allow for simple and robust inference. Gibbs sampling and variational Bayesian inference are simple with exponential conjugate family distributions. Robustness with respect to the actual choice of hyperparameters comes almost for free for large data sets because the likelihood will dominate over the prior. Explicit

expressions for the distributions used are given in Appendix B. We let each item factor have normal distribution:

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

with a Normal-Wishart priors

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

So the factors are conditional independent given the shared mean and covariance. We use a completely analogous model for the user factors. The hyperparameter of the Normal-Wishart $(\beta_0, \mathbf{W}_0, \nu_0)$ will have to be specified by the user. Our settings and the parametrization of the the distributions used are given in Appendix **??**. The inverse variance parameter γ is non-negative so a Gamma-distribution is a standard choice:

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

Summary of the model. The joint distribution of data and model parameters

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

is: using the definitions in equations (2), (3), (4), and (5)

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

Predictive distribution. **If r is on a numeric scale we can determine $\langle r \rangle$, else just the probability of r taking on a certain rank, e.g. A, B, etc.** The optimal point estimate from the model in a least squares sense is the expected prediction for $r = r_{mn}$:

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

Since the aim of the aim of the Netflix prize is to minimize the root mean squared error (RMSE) this is the best point estimate to use. The summation over r for constant parameters

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

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

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

This expression should then be averaged over $p(\mathbf{u}_m, \mathbf{v}_n, \gamma|\mathcal{D})$ in order to make predictions. Gibbs sampling and variational Bayes approaches to this is given in the two sections.

3. Gibbs sampling

Gibbs sampling amounts to sampling from the conditionals of the model sequentially. Under some mild conditions this will sample from the posterior distribution (Robert and Casella, 2004). **we can loosely mention these here...** We go over these in turn and summarize with pseudo-code for the sampler. Apart from the latent variables these are standard updates.

Factors.

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

where $\Omega(m)$ is the set of users for item m. A similar update exist for $\mathbf{v}_n$. The important point here is that in order to sample factor m we only need γ and the variables directly connected with m: $\{(\mathbf{v}_n, h_{mn}|n \in \Omega(m)\}$. This suggest an obvious order of the updates where we sample the latent variables when needed rather than storing them, see pseudo-code below.

Latent variables. Sampling the conditional for the latent variable

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

requires one evaluation of a unit interval random number, one random normal number, two evaluations of Φ and one of Φ^{-1}. It is possible to avoid the normal but that requires changing the likelihood function to a step function. To derive the sampler we introduce the "noise-free" latent variable

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

For any m and n, which we omit here for brevity, the joint marginal distribution of r, f, and h, given $\mu = \mathbf{u}\cdot\mathbf{v}$ and γ, is

$$p(r|f)\,p(f|h)\,p(h|\mu,\gamma) = \Big[\Theta(b_{r+1}-f) - \Theta(b_r - f)\Big]\,\mathcal{N}(f; h, 1)\,\mathcal{N}(h; \mu, \gamma^{-1})\ . \tag{6}$$

Density (6) can be sampled from in two steps, $f|r$ and $h|f$. The distribution $f|r$ is a truncated normal:

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

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

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

The desired sample is obtained from:

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

Algorithm 1 Gibbs sampling

initialize $\mathbf{U}$, $\mathbf{V}$, $\boldsymbol{\mu}_u$, $\boldsymbol{\Psi}_u$, $\boldsymbol{\mu}_v$, $\boldsymbol{\Psi}_v$
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: $\Delta_n = \sum_{m \in \Pi(n)} (h_{nm} - \mathbf{u}_m \cdot \mathbf{v}_n)^2$
  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, we sample for $\boldsymbol{\mu}_u$ and $\boldsymbol{\Psi}_u$ from

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

where

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

and can sample for $\boldsymbol{\mu}_v$ and $\boldsymbol{\Psi}_v$ in the same way. For the gamma distribution we have

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

Pseudo code The pseudo code is given in algorithm 1. Note that

- Using a Gaussian likelihood instead of ordinal regression simply amounts to replacing h_{mn} by r_{mn}.
- The variational Bayes algorithm given below will have exactly the same structure. All sampling steps are replaced with updates of sufficient statistics.
- An efficient data structure for this algorithm stores the data twice such that we can easily access the rating of the sets $\Omega(m)$ and $\Pi(n)$.

4. Variational Bayes

Variational Bayes aims at approximating $p(\theta|\mathcal{D})$ with a simpler distribution $q(\theta)$. The simpler distribution is typically a factorized distribution in the exponential family. We are really replacing a complicated Bayesian network with a simpler factorized one...

$q(\theta)$ is found my minimizing the variational free energy, and can be obtained through the VBEM algorithm **lots of details & ref**.

The factorized distribution is

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

Apart from latent variables all updates are standard. We will not give pseudo-code for this algorithm since it has exactly the same structure as for Gibbs sampling.

As we only need $\langle h_{nm}\rangle$ when updating the variational distributions $q(\mathbf{u}_m)$ and $q(\mathbf{v}_n)$, we don't need to define a distribution for it, but only determine its mean online...

We let $q(\mathbf{u}_m) = \mathcal{N}(\mathbf{u}_m; \langle\mathbf{u}_m\rangle, \boldsymbol{\Sigma}_n)$ and $q(\mathbf{v}_n) = \mathcal{N}(\mathbf{v}_n; \langle\mathbf{v}_n\rangle, \boldsymbol{\Xi}_n)$. The factorized approximations are also Normal-Wishart, parameterized with $\boldsymbol{\mu}_{u_{\text{VB}}}$, $\beta_{u_{\text{VB}}}$, $\mathbf{W}_{u_{\text{VB}}}$, and $\nu_{u_{\text{VB}}}$:

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

Latent variables.

$$\begin{aligned} q(h_{nm}) &\propto p(r_{nm}|h_{nm}) \exp\left(\langle \log p(h_{nm}|\boldsymbol{\lambda}_n, \mathbf{x}_m, \sigma^2)\rangle_{q(\boldsymbol{\lambda}_n)q(\mathbf{x}_m)}\right) \\ &\propto p(r_{nm}|h_{nm})\mathcal{N}(h_{nm}; \langle\boldsymbol{\lambda}_n\rangle^T\langle\mathbf{x}_m\rangle, \sigma^2) \\ \langle h_{mn}\rangle &= ... \end{aligned}$$

Factors The full update is:

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

$$\boldsymbol{\Sigma}_m = \left(\langle\boldsymbol{\Psi}_u\rangle + \gamma\sum_{n\in\Omega(m)}(\boldsymbol{\Xi}_n + \langle\mathbf{v}_n\rangle\langle\mathbf{v}_n^T\rangle)\right)^{-1},$$

where $\langle\boldsymbol{\Psi}_u\boldsymbol{\mu}_u\rangle = \langle\boldsymbol{\Psi}_u\rangle\langle\boldsymbol{\mu}_u\rangle = \nu_{u_{\text{VB}}}\mathbf{W}_{u_{\text{VB}}}\boldsymbol{\mu}_{u_{\text{VB}}}$.

Because of spatial constraints we only store a diagonal update, and constrain $\boldsymbol{\Sigma}_n$ and $\boldsymbol{\Xi}_n$ to be diagonal! For the Normal-Wishart prior the updates will look slightly different. We have $q(u_{mi}) = \mathcal{N}(u_{mi}; \langle u_{mi}\rangle, \Sigma_{mi})$, with

$$\Sigma_{mi} = \left(\langle\boldsymbol{\Psi}_u\rangle_{ii} + \gamma\sum_{n\in\Omega(m)}(\Xi_{ni} + \langle v_{ni}\rangle^2)\right)^{-1}$$

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

Note: $\langle \mathbf{\Psi}_u \rangle_i$ indicates column i of $\langle \mathbf{\Psi}_u \rangle$, and $\langle \mathbf{\Psi}_u \rangle_{ii}$ diagonal element (i,i). We also use $\langle v_{ni} \rangle \langle u_{mi} \rangle - \langle \mathbf{v} \rangle \cdot \langle \mathbf{u}_m \rangle$, which is equivalent to $-\sum_{j \neq i} \langle v_{nj} \rangle \langle u_{mj} \rangle$.

are we going to include gamma?

$$q(\gamma) = \cdots$$

Wishart update

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

5. Netflix prize

5.1 Hyperparameter settings for the Netflix prize.

For the Netflix problem we have ranks 1 to 5. We set the boundaries to be equidistant with steps $\Delta b = 4$: $(b_1, \ldots, b_6) = (-\infty, -6, -2, 2, 6, \infty)$. Any other equidistant choice would lead to the same result with appropriate re-scaling of the remaining parameters.

The actual setting of the hyperparameters of the Normal-Wishart prior for the mean and variance of the factors, $\mathcal{N}(\boldsymbol{\mu}_u; \boldsymbol{\mu}_0, (\beta_0 \mathbf{\Psi}_u)^{-1}) \mathcal{W}(\mathbf{\Psi}_u; \mathbf{W}_0, \nu_0)$ are not very important because there is so much data to learn from so any reasonably weak prior will by overridden by the data. To reflect this, we simply set the following values: the mean of the mean prior is $\boldsymbol{\mu}_0 = \mathbf{0}$, the mean-precision coupling parameter is set to one $\beta_0 = 1$, the precision matrix to the identity matrix $\mathbf{W} = \mathbf{I}$ and the degrees of freedom to the dimensionality $D = K$.

The most important parameter in the problem is γ, the inverse variance of the h latent variable distribution. This parameter will to a large extent determines the degree of fitting to the training data. We first used $a_0 = 10$ and $b_0 = 10^{-2}$ in the gamma-distribution prior corresponding to a prior 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 root mean squared deviation (RMSE) of $\sqrt{(\langle \gamma^{-1} \rangle}/\delta b = 3/4$. This value is well below the Netflix prize goal. When we ran with these parameter setting for $K = 50$, γ converged to something like 0.119 with very small fluctuations since it is well-determined by the large amount of data. It turned that keeping γ fixed at value $\sim 0.8 - 0.1$ gives somewhat better performance. We ascribe this to the fact that the training and qualifying data are not sampled from the entirely same distribution. The qualifying set contains more recent ratings and there has been a substantial upward drift in the ratings over time. We thus depart from a purely Bayesian approach for this parameter and have not looked into or tried to model this interesting aspect of the Netflix prize problem.

5.2 Performance results

6. Conclusion

Appendix A. Asymptotics of the error function

Appendix B. Standard distributions

We use the following standard distributions: Normal $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, Wishart $\mathcal{W}(\cdot; \mathbf{W}, \nu)$ with scale matrix $\mathbf{W}$ and ν degrees of freedom and the Gamma distribution $\Gamma(\cdot; a, b)$ with shape parameter a and scale (inverse rate) parameter b. The explicit expressions (up to a normalizing factor) for the Wishart and the Gamma are

$$\begin{aligned} \mathcal{W}(\boldsymbol{\Psi}; \mathbf{W}, \nu) &\propto |\boldsymbol{\Psi}|^{(\nu-D+1)/2} \exp(-\frac{1}{2}\mathrm{tr}[\mathbf{W}\boldsymbol{\Psi}]) \\ \Gamma(\gamma; a, b) &= \frac{1}{b^a \Gamma(a)} \gamma^{a-1} \exp\left(-\frac{\gamma}{b}\right) . \end{aligned}$$

References

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

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.

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 International Conference on Machine Learning*, volume 25, 2008.