Large-scale Ordinal Collaborative Filtering

Ulrich Paquet, Blaise Thomson, and Ole Winther

Microsoft Research Cambridge, University of Cambridge, Technical University of Denmark
ulripa@microsoft.com,brmt2@cam.ac.uk,owi@imm.dtu.dk

Abstract. This paper proposes a hierarchical probabilistic model for ordinal matrix factorization by actively modelling the ordinal nature of ranking data, which is typical of large-scale collaborative filtering tasks. Two algorithms are presented for inference, one based on Gibbs sampling and one based on variational Bayes. 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 improves similar factorization techniques. Results also show how Gibbs sampling outperforms variational Bayes on this task, despite the large number of ratings and model parameters.

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 of results, $\mathbf{R}$, where rows represent the observation number and columns represent the variables of interest. This matrix is then decomposed as $\mathbf{R} = \mathbf{U}^\top \mathbf{V} + \boldsymbol{\epsilon}$, 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 $\mathbf{R}$.

This paper discusses a model for matrix factorization when the observed data is a collection of ranks, also known as ordinal data. It contributes:

- An extended probabilistic model for ordinal matrix factorization. The additional parameters express a rich model, through which the prior distributions over factors are flexible, but remain coupled through shared hyper-priors. Salakhutdinov and Mnih's hierarchical framework of matrix factorization [7] is coupled with a likelihood function that models the ordinal nature of the data [2].
- An efficient variational approach for inference in the model.

– An efficient Gibbs sampling algorithm for inference in the model.

The model is tested on the Netflix data set, which is a collection of around 100 million movie ratings. The matrix is sparse in that many of the movie-user pairs have no rating. 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 improve or equal alternative matrix factorization approaches on this metric. There exists a rich ensemble of similar probabilistic factorization models for large-scale collaborative filtering [4, 7, 9]. Alternative approaches to collaborative filtering are equally common, such as those using user rating profiles [5], restricted Boltzmann machines [8], nearest neighbours [1], and non-parameteric models [10, 11]. The best performance on this data has been achieved by averaging ensembles of many different models [3].

2 Probabilistic model

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 ($r = 5$) is regarded as superior to one with a four-star rating ($r = 4$), which in turn is better than one with a three-star rating. Instead of directly modelling the ranks, $r = 1, \ldots, R$, we will model a collection of hidden variables, h. The probability distribution of the ranks will depend on the position of h compared to a collection of fixed boundaries b_r,

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

Let r_{mn} denote the rank for row $m = 1, \ldots, M$ and columns $n = 1, \ldots, N$, or *user n's rank for item m*. The probability of r_{mn} in terms of a hidden variable h_{mn} is

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

where $\Phi(x) = \int_{-\infty}^{x} \mathcal{N}(z; 0, 1)\, dz$ is the cumulative Gaussian density or probit function. The hidden variables, h_{mn}, are modelled by a Gaussian distribution, with mean equal to the dot product of a row factor $\mathbf{u}_m$ and a column factor $\mathbf{v}_n$,

$$p(h_{mn}|\mathbf{u}_m, \mathbf{v}_n, \gamma) = \mathcal{N}(h_{mn}\,;\, \mathbf{u}_m^\top \mathbf{v}_n,\, \gamma^{-1}) \ . \tag{2}$$

Factors $\mathbf{u}_m$ and $\mathbf{v}_n$ can be viewed as item m and user n's latent traits, each of which has a Normal prior distribution. The $\mathbf{u}_m$'s are conditionally independent given a shared mean and covariance,

$$p(\mathbf{u}_m|\boldsymbol{\mu}_u, \boldsymbol{\Psi}_u) = \mathcal{N}(\mathbf{u}_m; \boldsymbol{\mu}_u, \boldsymbol{\Psi}_u^{-1}) \ . \tag{3}$$

A completely analogous model is used for the user factors $\mathbf{v}_n$. The mean and precision matrix (inverse covariance) is modelled with a conjugate Normal-Wishart prior

$$p(\boldsymbol{\mu}_u, \boldsymbol{\Psi}_u) = \mathcal{N}(\boldsymbol{\mu}_u; \boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Psi}_u)^{-1})\, \mathcal{W}(\boldsymbol{\Psi}_u; \mathbf{W}_0, \nu_0) \ . \tag{4}$$

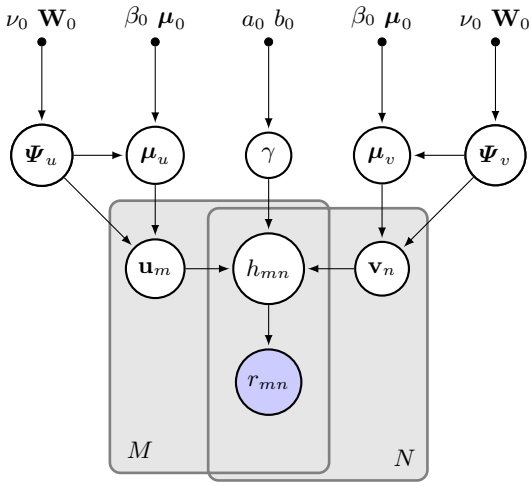

Fig. 1. Graphical model for factor model Bayesian hierarchy as it is described in the main text.

The Wishart distribution $\mathcal{W}(\boldsymbol{\Psi}; \mathbf{W}, \nu) \propto |\boldsymbol{\Psi}|^{(\nu-K+1)/2} \exp(-\frac{1}{2}\mathrm{tr}[\mathbf{W}^{-1}\boldsymbol{\Psi}])$ over symmetric positive definite matrices is parameterized with a scale matrix $\mathbf{W}$ and ν degrees of freedom. 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. Figure 1 summarizes the joint distribution of the data and model parameters $\theta = \{\mathbf{H}, \mathbf{U}, \mathbf{V}, \gamma, \boldsymbol{\mu}_u, \boldsymbol{\Psi}_u, \boldsymbol{\mu}_v, \boldsymbol{\Psi}_v\}$ in a Bayesian network. In Section 3 we will sample from, and in Section 4 approximate the posterior distribution $p(\theta|\mathcal{D})$.

3 Gibbs sampling

Gibbs sampling is a MCMC method that sequentially samples from the conditional distributions of the model. We briefly describe two sampling steps here.

Factors. With $\Omega(m)$ being the set of users that rated item m, the conditional distribution for each item factor $\mathbf{u}_m$ is Gaussian

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

with covariance $\boldsymbol{\Sigma}_m = (\boldsymbol{\Psi}_u + \gamma \sum_{n \in \Omega(m)} \mathbf{v}_n \mathbf{v}_n^\top)^{-1}$. Notice that the distribution for factor m only requires knowledge of γ and the variables directly connected with m: $\{(\mathbf{v}_n, h_{mn} | n \in \Omega(m)\}$.

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}. 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 is omitted here for brevity, the joint marginal distribution of r, f, and h, given $\mu = \mathbf{u}^\top \mathbf{v}$ and γ, is

$$p(r|f)\, p(f|h)\, p(h|\mu,\gamma) = \Big[\Theta(b_{r+1}-f) - \Theta(b_r - f)\Big]\, \mathcal{N}(f; h, 1)\, \mathcal{N}(h; \mu, \gamma^{-1})\ . \quad (5)$$

The density $f, h|r, \mu, \gamma$ in (5) 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

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

with $\Phi_{\max} = \Phi((b_{r+1}-\mu)/\sqrt{1+\gamma^{-1}})$ and $\Phi_{\min} = \Phi((b_r-\mu)/\sqrt{1+\gamma^{-1}}))$. A sample can be drawn from $p(f|r,\mu,\gamma)$ using the cumulative distribution $f = \mu + \sqrt{1+\gamma^{-1}}\ \Phi^{-1}(\Phi_{\min} + \mathrm{rand}\,(\Phi_{\max} - \Phi_{\min}))$, where "rand" gives a uniform random number between zero and its argument. The desired sample is obtained from $p(h|f,\mu,\gamma) = \mathcal{N}\left(h\,;\, (f+\gamma\mu)/(1+\gamma),\, (1+\gamma)^{-1}\right)$.

4 Variational Bayes

Variational Bayes (VB) is one popular algorithm for obtaining an approximation $q(\theta)$ to $p(\theta|\mathcal{D})$. 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. An approximating family

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

is chosen, with the item and user factors having Gaussian approximations $q(\mathbf{u}_m) = \mathcal{N}(\mathbf{u}_m; \langle\mathbf{u}_m\rangle, \boldsymbol{\Sigma}_m)$ and $q(\mathbf{v}_n) = \mathcal{N}(\mathbf{v}_n; \langle\mathbf{v}_n\rangle, \boldsymbol{\Xi}_n)$. The factorized approximations for the hierarchical parameters follow the prior's Normal-Wishart form, while $q(\gamma)$ is chosen to be a Gamma density. We present two example VB updates:

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

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

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

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

$$q(h_{mn}) \propto p(r_{mn}|h_{mn}) \ \exp\left(\left\langle \log p(h_{mn}|\mathbf{u}_m, \mathbf{v}_n, \gamma)\right\rangle_{q(\mathbf{u}_m)\, q(\mathbf{v}_n)\, q(\gamma)}\right) \ .$$

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 expression for $\langle h_{nm} \rangle$ is ($\langle h_{nm}^2 \rangle$ can be similarly evaluated)

$$\langle h_{mn} \rangle = \mu + \frac{\gamma^{-1}}{\sqrt{1+\gamma^{-1}}} \frac{\mathcal{N}(z_r) - \mathcal{N}(z_{r+1})}{\Phi(z_r) - \Phi(z_{r+1})} \ .$$

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 $\mathbf{R}$. A test or "qualifying" set of almost three million user–movie pairs for which the ratings are withheld. Algorithms are benchmarked by their RMSE computed over an unknown half of the test set.

Performance results We compare inference with Gibbs sampling and VB for a number latent dimensions, $K = 50, 100, 200$, and γ settings $\gamma = 0.8, 0.9, 0.10$, and one setting in which γ is also inferred. The results are summarized in Figure 2. 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 far more parameters than the data set size $|\mathcal{D}|$ is possible with Bayesian averaging [6].

6 Conclusion and outlook

This paper has proposed a hierarchical model for ordinal matrix factorization. Comparing our results to the regular factor model of [7], we see that the ordinal likelihood and hierarchical priors give substantial improvements (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). It is therefore clear that the use of ordinal likelihoods and hierarchical models is important when modeling ordinal data.

Two standard methods of inference were compared: Gibbs sampling and variational Bayes. The comparison of the two approaches on such a large task gives rise to several conclusions that are applicable to other similar settings (factor models with relatively high noise levels). Variational inference for the smaller factor sizes ($K = 50$) showed promising results, but unfortunately overfitted for

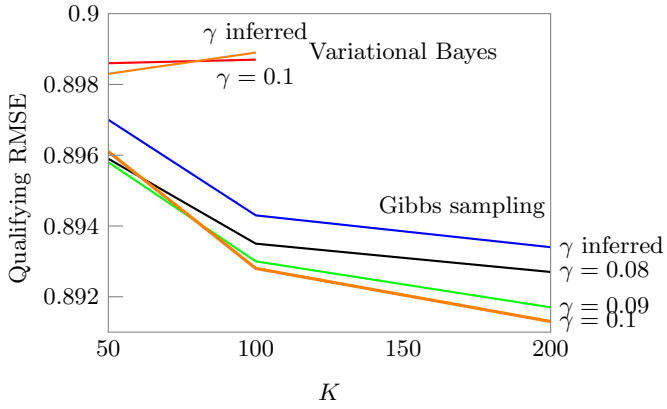

Fig. 2. RMSE on the qualifying set as a function of K for different γ-settings. The four lower lines are for Gibbs sampling and the two upper lines for VB.

larger models. On this problem, Gibbs sampling performed better and would generally be recommended.

In this paper a minimal model was used to keep the message as clean as possible. The results can clearly be improved by blending with other models, as was shown by [3]. Future work will evaluate the extent to which the gains obtained from this model affect the performance of a blended model.

References

1. Bell, R.M., Koren, Y.: Improved neighborhood-based collaborative filtering. In: Proceedings of KDD Cup and Workshop (2007)
2. Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. Journal of Machine Learning Research 6, 1019–1041 (2005)
3. Koren, Y.: The BellKor solution to the Netflix Grand Prize. Tech. rep. (2009)
4. Lim, Y.J., Teh, Y.W.: Variational Bayesian approach to movie rating prediction. In: Proceedings of KDD Cup and Workshop (2007)
5. Marlin, B.: Modeling user rating profiles for collaborative filtering. In: Thrun, S., Saul, L., Schölkopf, B. (eds.) NIPS (2004)
6. Neal, R.: Bayesian Learning for Neural Networks. Springer-Verlag (1996)
7. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In: ICML (2008)
8. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In: ICML (2007)
9. Stern, D.H., Herbrich, R., Graepel, T.: Matchbox: large scale online Bayesian recommendations. In: WWW. pp. 111–120 (2009)
10. Yu, K., Lafferty, J., Zhu, S., Gong, Y.: Large-scale collaborative prediction using a nonparametric random effects model. In: ICML (2009)
11. Zhu, S., Yu, K., Gong, Y.: Stochastic relational models for large-scale dyadic data using MCMC. In: NIPS (2009)