Bayesian Matrix Factorization with Side Information and Dirichlet Process Mixtures

Ian Porteous and **Arthur Asuncion** and **Max Welling**
Bren School of Information and Computer Science
University of California Irvine
Irvine, CA 92697
{iporteou,asuncion,welling}@ics.uci.edu

Abstract

Matrix factorization is a fundamental technique in machine learning that is applicable to collaborative filtering, information retrieval and many other areas. In collaborative filtering and many other tasks, the objective is to fill in missing elements of a sparse data matrix. One of the biggest challenges in this case is filling in a column or row of the matrix with very few observations. In this paper we introduce a Bayesian matrix factorization model that performs regression against side information known about the data in addition to the observations. The side information helps by adding observed entries to the factored matrices. We also introduce a nonparametric mixture model for the prior of the rows and columns of the factored matrices that gives a different regularization for each latent class. Besides providing a richer prior, the posterior distribution of mixture assignments reveals the latent classes. Using Gibbs sampling for inference, we apply our model to the Netflix Prize problem of predicting movie ratings given an incomplete user-movie ratings matrix. Incorporating rating information with gathered metadata information, our Bayesian approach outperforms other matrix factorization techniques even when using fewer dimensions.

Introduction

Matrix factorization is an important technique in machine learning which has proven to be effective for collaborative filtering (Koren 2008), information retrieval (Deerwester et al. 1990), image analysis (Lee and Seung 1999), and many other areas. A drawback of standard matrix factorization algorithms is that they are susceptible to overfitting on the training data and require careful tuning of the regularization parameters and the number of optimization steps. Bayesian approaches to matrix factorization (Salakhutdinov and Mnih 2008; Porteous, Bart, and Welling 2008; Blei, Ng, and Jordan 2003) attempt to address this weakness by integrating over model parameters and hyperparameters, thus allowing for complex models to be learned without requiring much parameter tuning.

Recent research using the Netflix Prize (Koren 2008) has shown that combining latent factor models with other models, such as neighborhood models, can improve performance. Although for the anonymized Netflix Prize data set there is not much additional information available for the customers, for many applications it is expected that additional side information about the customers or movies/products would be available and beneficial to incorporate into a model. Incorporating side information is particularly important when there are few observations, as in the case of a new movie or product.

We also introduce a Dirichlet process mixture of normal distributions as a prior for the matrix decomposition. The prior plays an important role in regularization when there are few observations for a row or column. In particular we expect that there are latent classes for the entities and regularization should be performed per class. Using the Netflix data, we confirm that latent classes can be discovered by examining the posterior distribution of the mixtures. We also evaluate our prediction performance on the Netflix Prize problem of predicting movie ratings and find that our approach is able to outperform other Bayesian and non-Bayesian matrix factorization techniques.

Our main contributions are the following: (1) We describe a general scheme for seamlessly incorporating known information; (2) We introduce a fully Bayesian nonparametric model which can be learned via a scalable Gibbs sampling algorithm; (3) We apply our approach to the Netflix Prize problem and are able to outperform many other factorization techniques without requiring tuning of parameters.

In the next section, we discuss related factorization approaches. We then describe our model in steps, first introducing the simpler related Bayesian probabilistic matrix factorization (Salakhutdinov and Mnih 2008), then adding side information and Dirichlet process mixture extensions to the model. Next, we present experiments showing the efficacy of our approach on the Netflix Prize problem. Finally, we conclude with a discussion of the applicability of our approach to more general machine learning problems.

For ease of explanation we will use movie ratings as an example throughout the paper and refer to movies and users. However, the model is general in form and not specialized for movie ratings or the Netflix competition.

Related work

Bayesian matrix factorization is a technique of growing interest. The work most closely related to our own is the Bayesian probabilistic matrix factorization model (Salakhut-

dinov and Mnih 2008), which features a Gaussian bi-linear factor model complete with Gaussian-Wishart priors. This model was applied to the Netflix problem and learned via Gibbs sampling. The Matchbox Bayesian recommendation system (Stern, Herbrich, and Graepel 2009) is another bilinear model featuring feedback and dynamics models and a similar mechanism to incorporate known information, with expectation propagation as the inference algorithm. (Agarwal and Chen 2009) introduce a model that also incorporates side information with latent factors. However, in their model the regression coefficients for the side information are treated separately from the latent factors. In our work the regression coefficients for side information are treated jointly with the rest of the latent factors. Variational Bayesian factorization methods have also been applied to the Netflix problem (Lim and Teh 2007).

On a more general level, nonparametric Bayesian factorization models such as those based on Indian Buffet Processes have been developed (Ghahramani, Griffiths, and Sollich 2007). While these models adaptively adjust the inner dimensionality of the matrix factorization, our model is nonparametric in the sense that the number of underlying data clusters can increase. Thus these techniques are potentially complementary to our approach.

Finally, there exist numerous non-Bayesian matrix factorization techniques, including variants of singular value decomposition, and these techniques have been successfully applied to the Netflix problem (Koren 2008; Takács et al. 2009). Later in the paper, we will show that our approach is competitive to both Bayesian and non-Bayesian techniques.

Model

First we will review BPMF which is a special case of our model. Reviewing BPMF will help to make our contributions clear. Next we will extend the model to include side information about movies, users or ratings. Finally, we introduce non-parametric prior distributions over latent feature vectors to complete the full model.

Bayesian probabilistic matrix factorization

Latent factor matrix factorization models for collaborative filtering assume that users and movies can be represented by vectors of latent factors U_i, V_j, where i and j designate the particular user and movie. Given the latent factor vectors for users and movies, a user's rating for a movie is predicted by the inner product of those vectors, $r_{ij} = U_i^T V_j$. In this way the matrix of all ratings R is factored into $U^T V = R$. The parameters of the model, are learned given the sparsely observed ratings matrix R.

BPMF (Salakhutdinov and Mnih 2008) puts matrix factorization in a Bayesian framework by assuming a generative probabilistic model for ratings with prior distributions over parameters. The full joint distribution of BPMF, $p(R, U, V, \Theta_u, \Theta_v | \sigma, \Theta_0)$ can be written as the product of

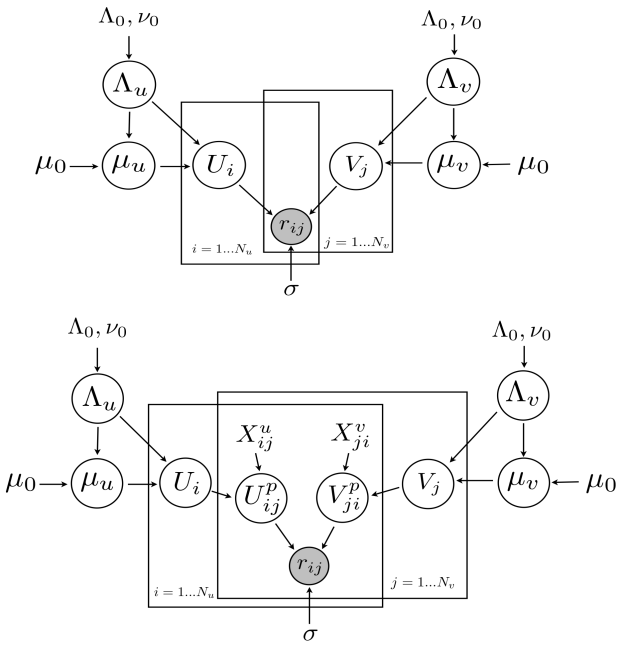

Figure 1: Top: Graphical model for Bayesian probabilistic matrix factorization (BPMF). Bottom: Graphical model for Bayesian matrix factorization with side information (BMFSI).

the following conditional distributions (Figure 1),

$$p(R|U, V, \sigma) = \prod_i \prod_j \left[\mathcal{N}(r_{ij}; U_i^T V_j, \sigma)\right]^{I_{ij}}, \quad (1)$$

$$p(U|\Theta_u) = \prod_i \mathcal{N}(U_i|\mu_u, \Lambda_u), \quad (2)$$

$$p(V|\Theta_v) = \prod_j \mathcal{N}(V_i|\mu_v, \Lambda_v), \quad (3)$$

$$p(\Theta_\kappa|\Theta_0) = \mathcal{N}(\mu_\kappa|\mu_0, \Lambda_\kappa/\beta_0)\mathcal{W}(\Lambda_\kappa|\Lambda_0, \nu_0)$$

where $\Theta_\kappa = \{\Lambda_\kappa, \mu_\kappa\}$, $\kappa = u, v$, and $\Theta_0 = \{\Lambda_0, \nu_0, \mu_0, \beta_0\}$. In words, BPMF has the following generative process:

1. For each user i sample a vector of parameters $U_i \sim \mathcal{N}(U_i|\mu_u, \Lambda_u)$,
2. For each movie j sample a vector of parameters $V_j \sim \mathcal{N}(V_j|\mu_v, \Lambda_v)$,
3. For each movie j rated by user i sample a rating $r_{ij} \sim \mathcal{N}(R_{ij}; U_i^T V_j, \sigma)$,

where the parameters of the multivariate normal distributions for parameter vectors, (Θ_v, Θ_u), are given a normal-Wishart prior. The posterior predictive distribution of a rating r_{ij} is found by marginalizing over model parameters, $\{U, V\}$ and hyperparameters, $\{\Theta_u, \Theta_v\}$:

$$p(r_{ij}|R^{\neg ij}, \Theta_0) = \iint p(r_{ij}|U_i, V_j) p(U, V|R^{\neg ij}, \Theta_u, \Theta_v) p(\Theta_u, \Theta_v|\Theta_0).$$

Since marginalizing over model parameters is analytically intractable, approximate inference using MCMC is performed.

Bayesian matrix factorization with side information

BPMF performs matrix factorization and prediction of new ratings based solely on the existing ratings. However, we would prefer a model that includes extra information if available. For example, if the user has explicitly told us they have a preference for a certain type of movie, we would want to incorporate that information into the model. As another motivating example outside movie recommendation, a bank may want to offer new products to its customers. However, besides monitoring the responses of customers to products in the past (similar to ratings) it also has an enormous amount of side information about those customers to exploit.

Next we show how it is possible to extend BPMF to include side information about movies, users or ratings. In the extended version, BPMF with side information (BMFSI), instead of just generating a rating from the product of latent factor vectors $U_i^T V_j$ we augment U, V with additional terms that contain information about the movie, user or rating. The augmented version of V_j is now specific to a rating, $V_{ji}^p = \{V_j, X_{ji}^v\}$. V_j contains the free parameters that will be learned for user j and X_{ji}^v contains additional side information about the rating against which the user i can regress. U_{ij}^p is similarly augmented.

We still calculate the predicted mean of each rating by taking the inner product of the augmented vectors U_{ij}^p and V_{ji}^p. To understand the consequences of this change we further segment U_{ij}^p and V_{ji}^p into three parts and examine how each part plays a role in calculating r_{ij}. The parts are depicted in table 1, and described below.

- The mean estimate of a rating is determined by the sum-product of the parts of the vectors U_{ij}^{pT} and V_{ji}^p: $\mu_{ij} = U_{ai}^T V_{aj} + U_{bi}^T X_{ji}^v + X_{ij}^{uT} V_{bj}$
- The first term, $U_{ai}^T V_{aj}$, is the matrix factorization term. If this is the only term, the model is BPMF.
- The second term, $U_{bi}^T X_{ji}^v$ is the result of user i's linear regression against the features of the movies they have rated or features of the rating itself. For example, if X_{ij}^v contains a flag indicating whether or not it is an action movie then the corresponding variable in U_{bi}^T indicates the user's bias towards action movies. X_{ij}^v can also contain rating-specific information, such as the date of the rating. In this case the corresponding variable in U_{bi}^T indicates the user's trend in ratings over time.
- The third term, $X_{ij}^{uT} V_{bj}$, is the complement to the second term and is the result of the movie's linear regression against features of the user or the rating-specific information. For example, just as in the second term, X_{ij}^{uT} could contain the date of the rating. The corresponding variable in the movie's vector V_{bj} indicates how the movie's ratings have trended over time. X_{ij}^{uT} could also contain information about the user who made the rating, such as whether or not the user is married. The movie can then learn a bias about how married users rate the movie.

U_{ij}^p	U_{ai}	U_{bi}	X_{ij}^u
V_{ji}^p	V_{aj}	X_{ji}^v	V_{bj}

Table 1: The rating for user i of movie j, r_{ij}, comes from the product of the augmented feature vectors U_{ij}^p, V_{ji}^p. In this table the feature vectors are arranged so that the sections of vectors that are multiplied together are adjacent to each other.

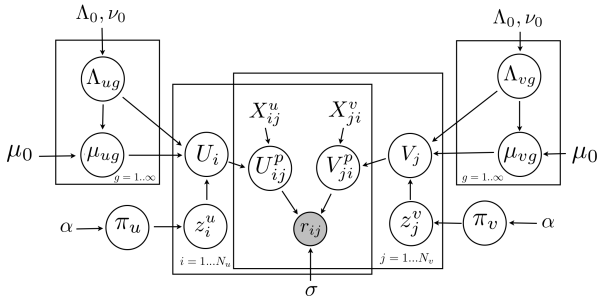

Figure 2: BMFSI with Dirichlet process mixtures

The model is very flexible as to what features are put into X_{ij}^u and X_{ji}^v. The features can be user-specific, movie-specific, or rating-specific. The only limitation is that they will be linearly combined in the final prediction. X_{ij}^u and X_{ji}^v can be the same size or different sizes. The feature vectors can be symmetric or different for users and movies.

The only change in the model specification is that U^p and V^p now replace U, V in the analogous BPMF equation 1, to give the following distribution for R

$$p(R|U^p, V^p, \sigma) = \prod_i \prod_j \left[\mathcal{N}(r_{ij}|U_i^{pT} V_j^p, \sigma).\right]^{I_{ij}} \quad (4)$$

The prior distributions for U_i, V_j remain the same as in BPMF equations 2 and 3 and are not replaced by U_i^p, V_j^p. However, the conditional distribution $p(U_i|\Theta_u, V^p, R)$ is different. In order to find the posterior distribution of a rating r_{ij} we again want to marginalize over the model parameters and hyperparameters.

Although the posterior distribution of r_{ij} is intractable as it is for BPMF, we can still derive an efficient Gibbs sampler to calculate an approximation. The Gibbs sampler works by cycling through the latent variables U, V and the parameters Θ_u, Θ_v, sampling each conditioned on the current values of all the other variables. The use of conjugate priors provides us with a convenient form for the conditional distributions of the latent variables.

BMFSI with Dirichlet process mixture prior

The BMFSI model assumes that every user and movie draws their vector of free parameters, U_i and V_j, from a single common multivariate normal distribution with a full covariance matrix. However, we expect that there are clusters of

movies or users that are more similar to each other then to the population in general. Consequently, a better generative model might be one where there are groups of users or movies and they draw their vector of latent factors from group specific distributions. In the generative process for this new model, instead of drawing a factor vector from a single common distribution, each user or movie first picks a group and then draws a vector from that group's distribution. So a user might first pick an action film group and then draw a vector of factors from the action group's distribution. To summarize, the model would have the following generative process:

1. For each user i, sample a group assignment $z_i^u \sim \pi_u$
2. For each movie j, sample a group assignment $z_j^v \sim \pi_v$
3. For each user i, sample a vector of parameters $U_i \sim \mathcal{N}(U_i|\mu_{z_i^u}, \Lambda_{z_i^u})$
4. For each movie j, sample a vector of parameters $V_j \sim \mathcal{N}(V_j|\mu_{z_j^v}, \Lambda_{z_j^v})$
5. For each movie j rated by user i, sample a rating $r_{ij} \sim \mathcal{N}(r_{ij}; U_i^{pT} V_j^p, \sigma)$

Since a priori we have no knowledge of the number of groups, we would like to use a non-parametric distribution that does not require us to specify the number of groups. To this end we use a Dirichlet process mixture (Antoniak 1974; Ferguson 1973) to model the user and movie latent feature vectors. The Dirichlet process mixture model has support for a countably infinite number of mixture components but only a few will dominate in the posterior, providing us with a convenient non-parametric distribution to work with. We will again want to marginalize over the parameters to find the posterior distribution of the rating predictions r_{ij}. Conditioned on the group assignment variables the Gibbs sampling algorithm is the same as the one for BMFSI, with the exception that there are per group Θ_{ug}, Θ_{vg} parameters to sample. Conditioned on the sampled value for U, V, sampling of the z^u, z^v is according to a Dirichlet process mixture model with U, V acting as data vectors. We use Algorithm 2 in (Neal 2000). Details of the sampling algorithm are provided in the supplementary material.

If we look at the samples of assignment variables from the posterior distribution of the model when applied to the Netflix prize data set we find support for our expectation that there are multiple distinct groups. In particular we find clusters of movies that are easily recognized as having common characteristics. We analyzed the assignment variables z from one sample of the Gibbs sampler after 200 iterations to inspect the clusters. For each of the 14 clusters found, we picked the top 5 largest movies by number of ratings and viewed their titles. The results from a qualitatively representative sample of the clusters are found in Table 3. The clusters of movies found by the model contain groups of movies for which we would expect the distribution of latent factors to be similar. There are also clusters of users, but they are more difficult to interpret because we lack labels.

Experiments

We evaluate our approach on the well-known Netflix Prize competition[1], an ideal collaborative filtering problem with a well-defined system for objectively comparing different solutions. The Netflix Prize data set is a sparse user-movie matrix of over 100,000,000 ratings where there are 480,189 users and 17,700 movies. Each rating is an integer from 1 to 5, and the date when the user rated the movie is also provided.

The objective of the competition is to predict ratings for a held-out portion of the data matrix, known as the Quiz set. Netflix also identifies a Probe set, a subset of the ratings in the given data matrix, which was generated using the same process that generated the Quiz set; thus, the Probe set is useful for internal evaluation. The evaluation metric used is the root mean squared error (RMSE) between the predicted ratings and the actual held-out ratings. To perform internal RMSE calculations for our experiments, we create a new Probe set consisting of 10% of the original Probe set, and we train on the rest of the given data (including the other 90% of the original Probe set). We also report results on the Quiz set.

We regress against the following side information in our experiments: the date (normalized from 0 to 1), a date flag (which indicates whether the rating's date was before March 12, 2004, where a change in the average rating was observed), the user's previous two ratings, and if available, the rating the user gave the K most similar movies measured by Pearson's correlation coefficients. We typically set $K = 5$. We also tried using other movie metadata which we extracted from Wikipedia (e.g. movie director, actors, languages) but we found that these Wikipedia-extracted features do not improve performance measured by RMSE.

The parameters that need to be manually set in our model are the number of user and movie dimensions D_u, D_v, and the σ, α, Θ_0 parameters (when $D_u = D_v$ we just use D to specify both). For all our experiments we set $\sigma = 0.8$. Λ_0 is set to the identity for all runs. $\mu_0 = 0$ for all runs. $\beta_0 = .8$ for all runs. We set $\alpha = .01$ for runs with side information and DP mixtures, $\alpha = .0000001$ for DP mixture runs without side information.

We ran our collapsed Gibbs samplers on a heterogeneous collection of machines, ranging from dual-core machines with 8GB RAM to 16-core machines with 128GB RAM. Parallelization across cores was achieved through OpenMP.

Since our approach is based on Gibbs sampling, it is important to average over many different samples in order to achieve good results. Figure 3(a) shows the Probe RMSEs for individual samples as well as the online average of samples during the sampler burn-in period, for various D. While each individual sample gives a fairly high RMSE (e.g. 0.93), averaging over samples from the posterior predictive distribution gives a substantially better RMSE (e.g. 0.89). In Figure 3(b), we run the sampler starting from the burn-in position and show the effect of averaging over multiple samples. As the number of samples approaches one hundred, there is little improvement from adding more samples.

[1] http://www.netflixprize.com

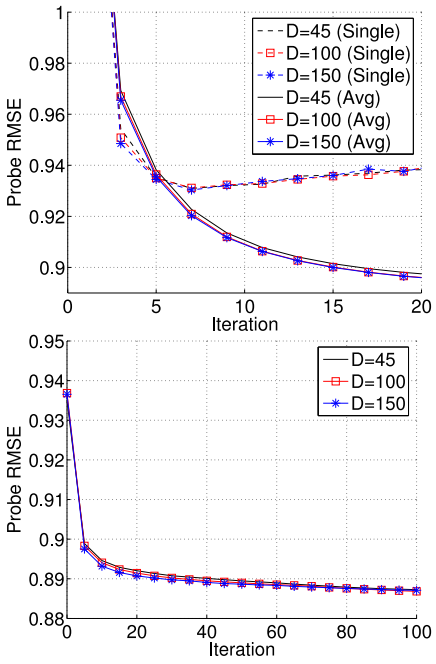

Figure 3: Top: Probe RMSEs during burn-in, for various D. Bottom: Probe RMSE after burn-in, with online averaging of samples.

D	Probe RMSE	Quiz RMSE
45	0.8970	–
45 (DP)	0.8957	0.8988
45 (SI)	0.8866	0.8909
45 (DP, SI)	0.8865	0.8907

Table 2: RMSEs for the model with/without side information (SI) and the Dirichlet process (DP).

We investigate the benefits of incorporating additional side information as well as activating the full nonparametric model. We perform runs with and without Dirichlet processes (DP), and with and without side information (SI). As shown in Table 2, a significant improvement is achieved when side information is included [2]. In Figure 4, we examine the mean of the coefficients (diagonal of Λ_u,Λ_v) for the side information dimensions and the collaborative filtering dimensions. In particular, the large coefficients in user dimensions 40-44 correspond to the side information for the five nearest neighbors, suggesting that this additional information plays a key factor in the model and in prediction.

Our results in Table 2 suggest that the multi-group model with Dirichlet process mixtures only marginally improves upon the single group model with side information for the Netflix prize data set. While this RMSE gain is insignificant for this data set, there are other benefits to having a Dirichlet process mixture model, such as the creation of clusters which are interpretable. After performing a hierarchical D=45 run, the sampler was able to find interesting movie clusters: a "Star Trek" cluster, an action movie cluster, a "Lord of the Rings" cluster, among many others. Table 3 shows the movie titles for four different clusters. The thematic cohesiveness within each cluster indicates that the model is able to learn coherent clusters of data points.

In summary, Table 4 shows the best RMSEs for competing factorization approaches. We find that our approach generally outperforms both Bayesian and non-Bayesian factorization techniques. We note that integrated models which combine matrix factorization with neighborhood models or time models can generally perform better than the factorization techniques below (Koren 2008). In addition, bagging results over many different methods can yield significantly better RMSEs, as evidenced by the final Netflix competition winner. However, we restrict our attention to matrix factorization and find that our approach is competitive with the other matrix factorization techniques.

[2]To ensure a fair comparison, whenever we include side information dimensions, we remove the equivalent number of free dimensions.

Twilight Zone: Vol. 16	Star Trek II: The Wrath of Khan
Twilight Zone: Vol. 22	Star Trek: Nemesis
Twilight Zone: Vol. 2	Star Trek: First Contact
Twilight Zone: Vol. 25	Planet of the Apes
Twilight Zone: Vol. 1	Star Trek: Insurrection
Sopranos: Season 1	Indiana Jones - Last Crusade
Sopranos: Season 2	The Matrix
Sopranos: Season 3	Raiders of the Lost Ark
South Park: Bigger,	Harry Potter - Chamber of Secrets
Sopranos: Season 4	The Matrix: Reloaded

Table 3: Largest movies by number of ratings for four different movie clusters.

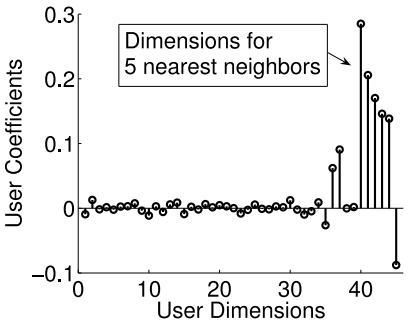

Figure 4: Coefficients learned on 45D run.

Discussion

We have shown that our Bayesian approach is competitive with other Bayesian and non-Bayesian factorization techniques. A benefit to our approach is that it can seamlessly handle additional information within the framework of a Bayesian factorization model, without requiring the tuning of many different parameters. Our results suggest that additional side information can act as a useful informative prior that can significantly improve results.

Method	Quiz RMSE	% Improvement
Cinematch Baseline	0.9514	0%
Variational Bayes (Lim and Teh 2007)	0.9141	3.73%
Matchbox (Stern, Herbrich, and Graepel 2009)	0.9100	4.14%
BPMF, D=60 (Salakhutdinov and Mnih 2008)	0.8989	5.25%
BPMF, D=300 (Salakhutdinov and Mnih 2008)	0.8954	5.60%
SVD++, D=50 (Koren 2008)	0.8952	5.62%
SVD++, D=200 (Koren 2008)	0.8911	6.03%
BRISMF D=250 (Takács et al. 2009)	0.8954	5.60%
BRISMF, D=1000 (Takács et al. 2009)	0.8904	6.10%
BMFSI (our model), D=45	0.8907	6.07%
BMFSI (our model), D=100	0.8875	6.39%

Table 4: Comparison between our model and other factorization techniques

The addition of Dirichlet process mixtures to the model provides a more flexible prior for the latent feature vectors and also discovers interpretable latent structure in the data. When used in combination with the side information it can provide a novel means of exploring groups in the data. For example, if the users are made up of married and un-married people with different preferences, then the model will likely form at least two groups for married and un-married users. Based on the mean latent factors of these groups, μ_{ug}, we could then examine group wide preferences for movies (or products) by plotting the inner-product of μ_{ug} with all the individual product vectors $\forall_j V_j$.

Possible extensions of our approach include using nonparametric techniques for adjusting the inner dimensionality D in tandem with our nonparametric approach over user clusters. Another facet of our approach that can be improved is the selection of side information to include.

In conclusion, we find that our Bayesian matrix factorization model that simultaneously performs regression on side information is scalable and produces accurate results on the Netflix Prize data. We believe that our factorization approach is applicable to problems in computer vision, and information retrieval, and adapting our approach to these domains remains future work.

Acknowledgments

This work is supported in part by NSF grants IIS-0447903 and IIS-0914783 as well as ONR/MURI grant 00014-06-1-073. Arthur Asuncion was supported by an NSF graduate fellowship.

References

Agarwal, D., and Chen, B.-C. 2009. Regression-based latent factor models. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 19–28. New York, NY, USA: ACM.

Antoniak, C. 1974. Mixtures of Dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics* 2:1152–1174.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Deerwester, S.; Dumais, S.; Furnas, G.; Landauer, T.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.

Ferguson, T. 1973. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics* 1:209–230.

Ghahramani, Z.; Griffiths, T.; and Sollich, P. 2007. Bayesian nonparametric latent feature models. *Bayesian Statistics* 8.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426–434. New York, NY, USA: ACM.

Lee, D., and Seung, H. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401:788–791.

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

Neal, R. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics* 9:283–297.

Porteous, I.; Bart, E.; and Welling, M. 2008. Multi-hdp: A non parametric bayesian model for tensor factorization. In *AAAI*, 1487–1490.

Salakhutdinov, R., and Mnih, A. 2008. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, 880–887. New York, NY, USA: ACM.

Stern, D.; Herbrich, R.; and Graepel, T. 2009. Matchbox: Large scale online bayesian recommendations. In *18th International World Wide Web Conference*, 111–111.

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