Collaborative Filtering via Rating Concentration

Bert Huang
Computer Science Department
Columbia University
New York, NY 10027
bert@cs.columbia.edu

Tony Jebara
Computer Science Department
Columbia University
New York, NY 10027
jebara@cs.columbia.edu

Abstract

While most popular collaborative filtering methods use low-rank matrix factorization and parametric density assumptions, this article proposes an approach based on distribution-free concentration inequalities. Using agnostic hierarchical sampling assumptions, functions of observed ratings are provably close to their expectations over query ratings, on average. A joint probability distribution over queries of interest is estimated using maximum entropy regularization. The distribution resides in a convex hull of allowable candidate distributions which satisfy concentration inequalities that stem from the sampling assumptions. The method accurately estimates rating distributions on synthetic and real data and is competitive with low rank and parametric methods which make more aggressive assumptions about the problem.

1 INTRODUCTION

This article proposes a novel approach to the collaborative filtering problem by exploiting the concentration of user and item statistics. By making a relatively agnostic assumption that users and items are drawn independently and identically-distributed (*iid*), it can be shown that the statistics of training ratings must concentrate close to the expectations of corresponding query ratings. Such assumptions are weaker than those used in previous approaches which often assume a parametric form on the generative model or assume that the rating matrix is low-rank. Nevertheless, an otherwise indifferent probability estimate that obeys such bounds (for instance the maximum entropy estimate) provides state-of-the-art performance.

The method described herein is largely complementary with current approaches since these leverage different intuitions, such as specific parametric forms for the distributions involved and low-rank constraints on the rating matrix. For instance, the assumption that the ratings matrix is low rank underlies many singular value decomposition (SVD) techniques. Therein, users and items are assumed to be *iid*, and ratings are assumed to be randomly revealed such that there is no bias between training and testing statistics. These assumptions already have been shown to be unrealistic (Marlin et al., 2007), however empirical performance remains promising. The SVD approach further assumes that the ratings are sampled from distributions parametrized by the inner products of low-dimensional descriptor vectors for each user and each item. In other words, the full rating matrix is assumed to be the product of a user population matrix and an item population matrix, possibly with additional independent noise. Often, such matrices are estimated with some form of regularization, either by truncating their rank, by penalizing their Frobenius norm or by placing Gaussian priors (a parametric assumption) on their descriptor vectors (Breese et al., 1998; Lim & Teh, 2007; Montanari et al., 2009; Rennie & Srebro, 2005; Salakhutdinov & Mnih, 2008b; Salakhutdinov & Mnih, 2008a; Srebro et al., 2005; Weimer et al., 2007).

Conversely, the approach in this article only makes minimal hierarchical sampling assumptions. It assumes that users and items are sampled *iid* and that each rating is subsequently sampled independently from a conditional probability distribution that depends on the respective user-item pair in an arbitrary manner. It is also assumed that the ratings are revealed randomly. The resulting learning algorithm makes no further assumptions about the distributions

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

or the interaction between items and users (such as the inner product assumption most low-rank matrix-factorization methods make). Subsequently, we prove that, as long as each rating distribution depends only on the user and item involved, statistics from a user's (or item's) training data concentrate around the expected averages of the query probabilities. The result is a concentration inequality which holds regardless of modeling assumptions. The combination of these concentration inequalities defines a convex hull of allowable distributions. With high probability, the desired solution lives within this set but is otherwise underdetermined. A reasonable way to select a particular member of this set is to identify the one that achieves maximum entropy (or minimum relative entropy to a prior). The maximum entropy criterion is merely used as an agnostic regularizer to handle the underdetermined estimation problem and ensure the uniqueness of the recovered estimate within the convex hull.

Since the dependencies in the rating process exhibit a hierarchical structure, the method proposed is reminiscent of the hierarchical maximum entropy framework (Dudík et al., 2007). In fact, the proposed algorithm can be viewed as a specific application of hierarchical maximum entropy where we estimate distributions linked by common parents (from which we have no samples) using statistics gathered from separate distributions with one sample each. Thus, the collaborative filtering setting is an extreme case of the hierarchical maximum entropy setup since, without the hierarchy, there would be no information about certain components of the probability model. Moreover, previous work (Dudík et al., 2007) proposed tree-structured hierarchies while this article explores a grid-structured (non-tree) hierarchy due to the matrix setup of users and items in the collaborative filtering problem.

We emphasize that the proposed intuitions and concentration inequalities herein complement previous parametric approaches and provide additional structure to the collaborative filtering problem. They may be used in conjunction with other assumptions such as low-rank matrix constraints. Similarly, the concentration bounds hold whether the data is generated by a distribution with known parametric form or by any arbitrary distribution.

2 ALGORITHM DESCRIPTION

Consider the collaborative filtering problem where the input is a partially observed rating matrix $X \in \mathbb{Z}^{M \times N}$. Each matrix element $x_{ij} \in \{1, \ldots, K\}$ is a random variable representing the rating provided by the i'th user for the j'th item where $i \in \{1, \ldots, M\}$ and $j \in \{1, \ldots, N\}$. The users $\{u_1, \ldots, u_M\}$ and the items $\{v_1, \ldots, v_N\}$ are variables drawn *iid* from arbitrary sample spaces $u_i \in \Omega_u$ and $v_j \in \Omega_v$, respectively. The observed ratings (where a sample of the random variable is provided) will be treated as a training set for the collaborative filtering problem and used to estimate unobserved ratings (where no sample of the random variable is available). The desired output is a set of predicted probability distributions on certain query ratings whose indices are specified a priori. Let T be the set of observed training (i, j) indices and let Q be the set of query indices. Given $\{x_{ij} | (i, j) \in T\}$ and Q, we wish to estimate the probabilities $\{p(x_{ij} | u_i, v_j) | (i, j) \in Q\}$.

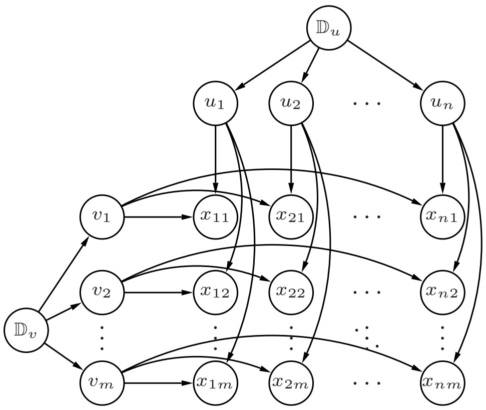

Figure 1: Graphical Model of Sampling Assumptions. We solve for the probabilities of the query ratings without explicitly estimating the user and item descriptors.

2.1 Assumptions

A major challenge in the sampling setting of collaborative filtering is that only a single sample rating[1] is observed from the training distributions $\{p(x_{ij} | u_i, v_j) | (i, j) \in T\}$ and *zero* samples are observed from the query distributions $\{p(x_{ij} | u_i, v_j) | (i, j) \in Q\}$. To transfer information from training samples to the query distributions, it will be helpful to make a hierarchical sampling assumption. Figure 1 depicts a graphical model representation of the proposed structure that will be used. First, users are drawn *iid* from an unknown distribution $p(u)$ and items are drawn *iid* from another unknown distribution $p(v)$. Subsequently, for some pairs of users and items, a rating is drawn independently with dependence on the corresponding item and user samples (the rating's parents in the graphical model).

It is natural to require that the ratings are samples from multinomial distributions over a range of rat-

[1] Recommendation data sets may include multiple ratings per user-item pair, though these are rare in practice.

ing values. In most collaborative filtering data sets (e.g., the Movielens data sets), ratings are discrete integer values (e.g., 1 to 5), so the multinomial is nonrestrictive. We further assume that the multinomial distributions are conditioned on a latent user descriptor variable $u_i \in \Omega_u$ and a latent item descriptor variable $v_j \in \Omega_v$ for each query $x_{ij} \in \{1, \dots, K\}$. In other words, we assume rating samples are drawn from $p(x_{ij}|u_i, v_j) = g(x_{ij}, u_i, v_j)$, where g provides an arbitrary mapping of the user and item descriptor variables to a valid multinomial distribution in the probability simplex. This is in contrast to standard (SVD) assumptions, which require that the function is constrained to be $g(x_{ij}, u_i^\top v_j)$, where the function g may be constrained parametrically and must depend solely on the inner product of low-dimensional *vector* descriptors in a Euclidean space.

Ratings x_{ij} for different users and items in this formulation are not identically distributed, not even for any particular user or item. The distribution $p(x_{ij}|u_i, v_j)$ for user i for an item j can be dramatically different from $p(x_{ik}|u_i, v_k)$, user i's rating distribution for another item $k \neq j$. However, the sampling structure in Figure 1 allows the transfer of information across many distributions since users (and items) are sampled *iid* from a common distribution $p(u)$ (and $p(v)$). The joint probability distribution implied by the figure factorizes as $\prod_{ij} p(x_{ij}|u_i, v_j)p(u_i)p(v_j)$ and, in particular, we are interested in recovering $\prod_{(i,j)\in Q} p(x_{ij}|u_i, v_j)$.

The aforementioned sampling assumptions will establish that the empirical average of any function of the ratings generated by a single user (or item) is close to its expectation with high probability. More specifically, empirical averages over training samples are close to corresponding averages over the expected values of the query distributions.

2.2 Concentration Bound

In this section, we present a theorem proving the concentration of training statistics to expected query averages. Specifically, we consider bounded scalar functions $f_k(x) \mapsto [0, 1]$ that take ratings as input and output a value inclusively between 0 and 1. Examples of such functions include the normalized rating itself (e.g., $(x-1)/(K-1)$, for ratings from 1 to K), or indicator functions for each possible value (e.g., $I(x = 1)$).

We will consider bounding the difference of two quantities. The first quantity is the empirical average of function $f_k(x)$ over the training ratings. Since this quantity is fixed throughout learning, we simplify notation by using μ_{ik} to denote this average of function $f_k(x)$ for user i's ratings, and using ν_{jk} to denote the average for item j's ratings. Let m_i be the number of training ratings for user i and let n_j be the number of training ratings for item j. These averages are then

$$\mu_{ik} = \frac{1}{m_i} \sum_{j|(i,j)\in T} f_k(x_{ij}), \;\; \nu_{jk} = \frac{1}{n_j} \sum_{i|(i,j)\in T} f_k(x_{ij}). \quad (1)$$

The second quantity of interest is the expected average of $f_k(x)$ evaluated on the query ratings. Let $\hat{m}_i$ be the number of query ratings for user i and $\hat{n}_j$ be the number of query ratings for item j. The expected averages are then expressed as

$$\begin{aligned} &\frac{1}{\hat{m}_i} \sum_{j|(i,j)\in Q} \mathbb{E}_{p(x_{ij}|u_i,v_j)}[f_k(x_{ij})], \\ &\frac{1}{\hat{n}_j} \sum_{i|(i,j)\in Q} \mathbb{E}_{p(x_{ij}|u_i,v_j)}[f_k(x_{ij})]. \end{aligned} \quad (2)$$

The following theorem bounds the differences between the quantities in Equation (1) and Equation (2).

Theorem 1. *For the ratings of user i, the difference*

$$\begin{aligned} \epsilon_{ik} &= \frac{1}{m_i} \sum_{j|(i,j)\in T} f_k(x_{ij}) \\ &\quad - \frac{1}{\hat{m}_i} \sum_{j|(i,j)\in Q} \mathbb{E}_{p(x_{ij}|u_i,v_j)}[f_k(x_{ij})] \end{aligned} \quad (3)$$

between the average of $f_k(x) \mapsto [0, 1]$ over the observed ratings and the average of the expected value of $f_k(x)$ over the query ratings is bounded above by

$$\epsilon_{ik} \leq \sqrt{\frac{\ln \frac{2}{\delta}}{2m_i}} + \sqrt{\frac{(m_i + \hat{m}_i) \ln \frac{2}{\delta}}{2m_i\hat{m}_i}} \quad (4)$$

with probability $1 - \delta$.

The proof is deferred to Appendix A. The same difference is also bounded by the following corollary.

Corollary 2. *The difference ϵ_{ik} defined in Equation (3) is bounded below by*

$$\epsilon_{ik} \geq -\sqrt{\frac{\ln \frac{2}{\delta}}{2m_i}} - \sqrt{\frac{(m_i + \hat{m}_i) \ln \frac{2}{\delta}}{2m_i\hat{m}_i}} \quad (5)$$

with probability $1 - \delta$.

Since Theorem 1 holds for any bounded function, applying the result for function $1-f_k(x)$ proves the corollary. Moreover, the same bounds hold for item ratings as summarized by the following corollary.

Corollary 3. *For the ratings of item j, the difference between the average of $f_k(x) \mapsto [0, 1]$ over the observed ratings and the average of the expected value of $f_k(x)$ over the query ratings is bounded above and below with high probability.*

The proof follows by replacing all references to users with references to items and vice versa. This produces concentration bounds that are similar to those in Equations (4) and (5).

Not only does Theorem 1 provide assurance that we should predict distributions with similar averages across training and query entries of X, the dependence of the bounds on the number of training and query samples adaptively determine how much deviation can be allowed between these averages. Due to the linearity of the expectation operator, each bound above produces a linear inequality or half-space constraint on $p(x_{ij}|u_i, v_j)$. The conjunction of all such half-spaces forms a convex hull Δ of allowable choices for the distribution $\prod_{(i,j)\in Q} p(x_{ij}|u_i, v_j)$. These bounds hold without parametric assumptions about the conditional probabilities $p(x_{ij}|u_i, v_j)$ generating the ratings. They also make no parametric assumptions about the distributions $p(u)$ and $p(v)$ generating the users and the items. The δ confidence value adjusts all the deviation inequalities and can be used as a regularization parameter. A small value of δ effectively relaxes the convex hull of inequalities while a large value of δ permits less deviation and shrinks the hull. Thus, δ controls all deviation inequalities which also individually depend on the cardinality of their training and query ratings.

2.3 Maximum Entropy

To choose from the candidate distributions $p \in \Delta$ that fit the constraints derived in the previous section (and reside inside the prescribed convex hull), we apply the maximum entropy method. The solution distribution p recovered will be of the form $\prod_{(i,j)\in Q} p(x_{ij}|u_i, v_j)$. We choose the distribution that contains the least information subject to the deviation constraints from the training data. Alternatively, we can minimize relative entropy to a prior p_0 subject to the constraints defined by the deviation bounds. This is a strictly more general approach since, when p_0 is uniform, minimum relative entropy coincides with standard maximum entropy. We suggest using a single identical maximum likelihood multinomial distribution over all ratings independent of user and item for the prior, namely $\prod_{(i,j)\in Q} p_0(x_{ij})$. Hence, we use the terms maximum entropy and minimum relative entropy interchangeably.

Assume we are given a set of functions F where $f_k \in F$ and $k \in \{1, \ldots, |F|\}$. Let α_i be the maximum deviation allowed for each function's average expected value for i. Let β_j be the maximum deviation allowed for each function's average expected value for item j. The α and β ranges are set according to Theorem 1 and its corollaries. For some δ, the allowed deviations are

$$\begin{aligned}\alpha_i &= \sqrt{\frac{\ln\frac{2}{\delta}}{2m_i}} + \sqrt{\frac{(m_i+\hat{m}_i)\ln\frac{2}{\delta}}{2m_i\hat{m}_i}}\\ \beta_j &= \sqrt{\frac{\ln\frac{2}{\delta}}{2n_j}} + \sqrt{\frac{(n_j+\hat{n}_j)\ln\frac{2}{\delta}}{2n_j\hat{n}_j}}.\end{aligned}$$

These scalars summarize the convex hull Δ of distributions that are structured according to the prescribed sampling hierarchy.

The primal maximum entropy problem is

$$\begin{aligned}&\max_p \sum_{ij\in Q} H(p_{ij}(x_{ij})) + \sum_{ij\in Q, x_{ij}} p_{ij}(x_{ij}) \ln p_0(x_{ij})\\ &\text{s.t.} \left|\frac{1}{\hat{m}_i}\sum_{j|ij\in Q}\sum_{x_{ij}} p_{ij}(x_{ij}) f_k(x_{ij}) - \mu_{ik}\right| \leq \alpha_i, \forall i, k\\ &\quad\ \left|\frac{1}{\hat{n}_j}\sum_{i|ij\in Q}\sum_{x_{ij}} p_{ij}(x_{ij}) f_k(x_{ij}) - \nu_{jk}\right| \leq \beta_j, \forall j, k.\end{aligned}$$

In the above, $p_{ij}(x_{ij})$ is used as shorthand for the conditional probabilities $p(x_{ij}|u_i, v_j)$ for space reasons. In practice, the dual form of the problem is solved. This is advantageous because the number of queries is typically $O(MN)$, for M users and N items, whereas the number of constraints is $O(M+N)$. Moreover, only a sparse set of the constraints is typically active. Since the primal problem is more intuitive, the details of the dual formulation are deferred to Appendix B. Given a choice of the feature functions F, a setting of δ and a set of observations, it is now straightforward to solve the above maximum entropy problem and obtain a solution distribution $\prod_{(i,j)\in Q} p(x_{ij}|u_i, v_j)$.

2.3.1 Feature Functions

This subsection specifies some possible choices for the set of feature functions F. These are provided only for illustrative purposes since many other choices are possible as long as the functions have $[0, 1]$ range. For discrete ratings $\{1, \ldots, K\}$, a plausible choice of F is the set of all possible conjunctions over the K settings. For example, the set of all singleton indicator functions and the set of pairwise conjunctions encodes a reasonable set of features when K is small:

$$\begin{aligned}f_i(x) &= I(x = i), & i &\in \{1, \ldots, K\},\\ f_{i,j}(x) &= I(x = i \vee x = j), & (i,j) &\in \{1, \ldots, K\}^2.\end{aligned}$$

These are used to populate the set F as well as the linear and quadratic transformation functions $f(x) = (x-1)/(K-1)$ and $f(x) = (x-1)^2/(K-1)^2$. Each of these functions is bounded by $[0, 1]$. It is thus possible to directly apply Theorem 1 and produce the constraints for the maximum entropy problem.

3 EXPERIMENTS

This section compares the maximum entropy (Maxent) method against other approaches for both synthetic and real data sets. A popular contender method is Fast Max-Margin Matrix Factorization (fMMMF) (Srebro et al., 2005) which factorizes the rating matrix subject to a low trace-norm prior. One variant of MMMF uses logistic loss to penalize training rating errors, which provides a smooth approximation of hinge-loss. The cost function for Fast MMMF with all-threshold logistic loss is as follows (Rennie, 2007):

$$||U||_{\mathrm{F}}^2 + ||V||_{\mathrm{F}}^2 + C \sum_{r,(i,j)\in T} \ln\left(1 + e^{\mathrm{sgn}(x_{ij}-r)(\theta_{ir} - u_i^\top v_j)}\right).$$

The sgn function outputs +1 when the input is positive and −1 otherwise. Consider a probabilistic interpretation of fMMMF where the above cost function is viewed as a log-loss associated with the likelihood

$$p(X, U, V|\theta) = \prod_{(i,j)\in T} p(x_{ij}|u_i, v_j, \theta) \prod_i p(u_i) \prod_j p(v_j).$$

The priors $p(u_i)$ and $p(v_j)$ on the user and item descriptors are zero-mean, spherical Gaussians scaled by $\frac{1}{C}$ and the conditional rating probability is defined by

$$p(x_{ij}|u_i, v_j, \theta) \propto \prod_r \frac{1}{1 + e^{\left(\mathrm{sgn}(x_{ij}-r)(\theta_{ir} - u_i^\top v_j)\right)}}. \quad (6)$$

The above allows direct comparison of log-likelihood performance of distributions estimated via Equation (6) versus the proposed maximum entropy method. The logistic-loss Fast MMMF method is evaluated using the author's publicly available code (Rennie, 2007).

Two additional comparison methods were considered: the Probabilistic Matrix Factorization (PMF) technique (Salakhutdinov & Mnih, 2008b) and its Bayesian extension (Salakhutdinov & Mnih, 2008a). Both methods learn the parameters of the graphical model structure in Figure 1. However, each makes parametric assumptions: Gaussian observation noise and Gaussian priors on the user and item descriptors. PMF performs *maximum a posteriori* (MAP) estimation on the model and Bayesian PMF uses Gibbs sampling to simulate integration over the Gaussian priors. Since PMF only estimates Gaussian means, the noise parameters can be set subsequently by choosing the value that produces the highest likelihood.

Finally, for experiments with real data, we also compare against the likelihoods using simple estimators such as a uniform distribution over all ratings or an identical maximum likelihood estimate $p_0(x_{ij})$ for all query ratings.

Table 1: Average Likelihood and Divergence Results for Synthetic Data. Values are averages over 10 folds and statistically significant improvements (according to a two-sample t-test) are displayed in bold. Higher log-likelihood and lower KL-divergence are better.

	fMMMF	Maxent
Log-Likelihood	-39690 ± 214	$\mathbf{-35732 \pm 216}$
KL-divergence	11254 ± 315	$\mathbf{4954 \pm 154}$

3.1 Synthetic experiments

One drawback of real data experiments is that ground truth rating distributions are not given, only samples from these are available. Therefore, consider a synthetic scenario where the exact rating distributions are specified and used to generate samples to populate the rating matrix.

First, 500 users and 500 items were sampled from a uniform probability density such that $u_i, v_j \in [0, 1]^5$. The rating distribution is then a multinomial with entries proportional to the element-wise product of the corresponding user-item pair: $p(x_{ij} = r|u_i, v_j) \propto u_i(r)v_j(r)$. Subsequently, a random subset T of training ratings was formed by drawing one sample from 20% of the entries of the rating matrix. These observed rating samples were provided to both fMMMF and Maxent. For both algorithms, 10% of the training set was used for cross-validation. The appropriate scalar regularization parameter (C or δ) was chosen by maximizing likelihood on this validation set.

A random subset Q of testing ratings was then formed by drawing one sample from 20% of the entries of the rating matrix (these entries were disjoint from the training entries). The out-of-sample test likelihood was then computed over Q. This setting is similar to a typical testing procedure with real data. Higher likelihood scores indicate a better estimate of the rating distribution however each rating distribution is only sampled once. Therefore, we also report the Kullback-

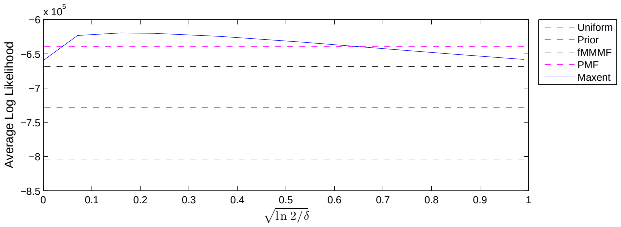

Figure 2: Average Log Likelihoods for each Algorithm on Movielens Data. The log-likelihoods are plotted against the regularization or confidence parameter δ in the maximum entropy method.

Leibler (KL) divergence between the true query multinomials and the estimated distributions $p(x_{ij}|u_i, v_j)$. The above experiment is repeated ten times and average results are reported across the trials. The maximum entropy method obtains higher likelihood scores as well as lower divergence from the true distribution than the fMMMF method. The advantages are statistically significant under both performance metrics. Table 1 summarizes the synthetic experiments.

3.2 Movie Ratings

This section compares several algorithms on the the popular Movielens data set. This data set is a collection of over one million ratings from over six thousand users for over three thousand movies. The ratings are integers ranging from 1 to 5. We randomly split the ratings in half to define the training and query user-item pairs. To choose regularization parameters, 20% of the training ratings were held out as a validation set. Finally, we test the resulting likelihoods on the query ratings.

On three random splits of training/validation/testing sets, the log-likelihood of the testing ratings was obtained for several methods. The maximum entropy method obtains the highest test likelihood, which improves over 20% more than the improvement obtained by the leading contender method relative to the naive $p_0(x)$ prior. Figure 2 illustrates the average likelihood for various regularization parameter settings compared to the competing methods. Our likelihood improvement is statistically significant according to a two-sample t-test with the rejection threshold below $1\mathrm{e}-5$. Log-likelihoods of each method are listed in Table 2.

We also compare ℓ_2 error on the ratings. This is done by recovering point-estimates for the ratings by taking the expected value of the rating distributions. Comparing against other maximum-likelihood methods like fMMMF and PMF, Maxent obtains slightly higher accuracy. All three methods are surpassed by the performance of Bayesian PMF, however. Interestingly, simply averaging the predictions of fMMMF and Maxent or the predictions of PMF and Maxent produces more accurate predictions than either algorithm alone. This suggests that the methods are complementary and address different aspects of the collaborative filtering problem. The ℓ_2 errors are listed in Table 3.

4 DISCUSSION

A method for collaborative filtering was provided that exploits concentration guarantees for functions of ratings. The method makes minimal assumptions about the sampling structure while otherwise remaining agnostic about the parametric form of the generative model. By assuming that users and items are sampled *iid*, general concentration inequalities on feature functions of the ratings were obtained. A solution probability distribution constrained by these concentration inequalities was obtained via the maximum entropy criterion. This method produced state-of-the-art performance by exploiting different intuitions and simpler assumptions than leading contenders. Furthermore, the proposed method is complementary with the assumptions in other approaches. Simply exploiting concentration constraints produces strong collaborative filtering results and more sophisticated models may also benefit from such bounds.

Acknowledgments

The authors thank J. Rennie, R. Salakhutdinov, and the anonymous reviewers and acknowledge support from the NetTrailMix Google Research Award.

References

Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of UAI 1998*.

Dudík, M., Blei, D., & Schapire, R. (2007). Hierarchical maximum entropy density estimation. *Proceedings of the ICML 2007* (pp. 249–256). Omnipress.

Huang, B., & Jebara, T. (2009). Exact graph structure estimation with degree priors. *Proceedings of ICMLA*.

Lim, Y. J., & Teh, Y. W. (2007). Variational bayesian approach to movie rating prediction. *Proceedings of KDD Cup and Workshop*.

Marlin, B., Zemel, R., Roweis, S., & Slaney, M. (2007). Collaborative filtering and the missing at random assumption. *Proceedings of UAI 2007*.

McDiarmid, C. (1989). On the method of bounded differences. *Surveys in Combinatorics*, 148–188.

Montanari, A., Keshavan, R., & Oh, S. (2009). Matrix completion from a few entries. *Proceedings of ISIT 2009*.

Rennie, J. (2007). *Extracting information from informal communication*. Doctoral dissertation, MIT.

Rennie, J., & Srebro, N. (2005). Fast maximum margin matrix factorization for collaborative prediction. *Proceedings of ICML 2005* (pp. 713–719).

Salakhutdinov, R., & Mnih, A. (2008a). Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. *Proceedings of ICML 2008*.

Salakhutdinov, R., & Mnih, A. (2008b). Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*.

Srebro, N., Rennie, J., & Jaakkola, T. (2005). Maximum-margin matrix factorization. *Advances in Neural Information Processing Systems 17*.

Table 2: Query Rating Log-Likelihoods on Movielens Data. The Maxent method has statistically significantly higher average likelihood over the three trials according to a two-sample t-test with p value of as little as $1\text{e}-5$. We convert the threshold model of fMMMF to a distribution for this comparison.

	Uniform	Prior	fMMMF Distrib.	PMF	Maxent
Split 1	-8.0489e+05	-7.2800e+05	-6.6907e+05	-6.3904e+05	-6.1952e+05
Split 2	-8.0489e+05	-7.2796e+05	-6.6859e+05	-6.3936e+05	-6.1977e+05
Split 3	-8.0489e+05	-7.2809e+05	-6.6819e+05	-6.3987e+05	-6.1931e+05
Average	-8.0489e+05	-7.2802e+05	-6.6862e+05	-6.3942e+05	**-6.1953e+05**

Table 3: Root Mean Square or ℓ_2 Performance of Various Algorithms. Maxent gives the least error among the MAP methods (fMMMF, PMF and Maxent) but Bayesian PMF outperforms all methods. Combining Maxent with other MAP methods improves accuracy.

	fMMMF	PMF	Maxent	BPMF	Maxent+fMMMF	Maxent+PMF
Split 1	0.9585	0.9166	0.9168	0.8717	0.9079	0.8963
Split 2	0.9559	0.9175	0.9162	0.8710	0.9052	0.8965
Split 3	0.9583	0.9186	0.9166	0.8723	0.9065	0.8973
Average	0.9575	0.9176	0.9165	0.8717	0.9065	0.8967

Weimer, M., Karatzoglou, A., Le, Q., & Smola, A. (2007). COFI RANK - maximum margin matrix factorization for collaborative ranking. *Advances in Neural Information Processing Systems 20*.

Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. (1997). Algorithm 78: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. *ACM Transactions on Mathematical Software*, *23*, 550–560.

A Concentration Proof

Proof of Theorem 1. The main intuition is that the total deviation between the expected query average and training average is composed of (1) the deviation between the training average and the expected training average and (2) the deviation between the expected training average and the expected query average. Since each component involves independent (though not necessarily *iid*) variables, McDiarmid's inequality (McDiarmid, 1989) can be invoked.

Recall the deviation of interest,

$$\begin{aligned}\epsilon_{ik} &= \frac{1}{m_i}\sum_{j|(i,j)\in T} f_k(x_{ij}) \\ &\quad -\frac{1}{\hat{m}_i}\sum_{j|(i,j)\in Q}\sum_{x_{ij}} f_k(x_{ij})p(x_{ij}|u_i,v_j).\end{aligned}$$

Clearly, ϵ_{ik} is a function of the independent ratings x_{ij} for all j such that $(i,j)\in T$ and a function of the independent item descriptors v_j for all j such that $(i,j)\in Q$. The Lipschitz constants of this function of two sets of independent variables will be examined. Since the range of function f_k is bounded by $[0,1]$, the deviation function ϵ_{ik} is Lipschitz continuous with constants $1/m_i$ for the training ratings and $1/\hat{m}_i$ for the query item variables. Furthermore, ϵ_{ik} is a function of two sets of independent variables allowing the application of McDiarmid's inequality (twice). After simplifying, the probability of ϵ_{ik} exceeding its expected value by a constant t_1 is bounded by

$$p\left(\epsilon_{ik}-\mathbb{E}_{x,v}[\epsilon_{ik}]\geq t_1\right)\leq\exp\left(-\frac{2m_i\hat{m}_i t_1^2}{m_i+\hat{m}_i}\right). \qquad (7)$$

Here, we write $\mathbb{E}_{x,v}$ to denote the expectation over the training ratings $\{x_{ij}|\ (i,j)\in\ T\}$ and all item descriptors, $\{v_j|(i,j)\in T\cup Q\}$. The expectation $\mathbb{E}[\epsilon_{ik}]$ is not exactly zero but can be shown to be close to zero with high probability. First, simplify the quantity using the linearity of expectation to obtain

$$\begin{aligned}\mathbb{E}_{x,v}[\epsilon_{ik}] &= \frac{1}{m_i}\sum_{j|(i,j)\in T}\mathbb{E}_x[f_k(x_{ij})]- \\ &\quad \frac{1}{\hat{m}_i}\sum_{j|(i,j)\in Q}\mathbb{E}_v\left[\sum_x f_k(x_{ij})p(x|u_i,v_j)\right].\end{aligned}$$

Rewrite the training expectation directly in terms of the training probabilities $\sum_{x_{ij}} f_k(x_{ij})p(x_{ij}|u_i,v_j)$. Similarly, since all the v variables are sampled *iid*, rewrite their expectation explicitly as follows

$$\begin{aligned}\mathbb{E}_{x,v}[\epsilon_{ik}] &= \frac{1}{m_i}\sum_{j|(i,j)\in T}\sum_{x_{ij}} f_k(x_{ij})p(x_{ij}|u_i,v_j)- \\ &\quad \frac{1}{\hat{m}_i}\sum_{j|(i,j)\in Q}\int_v\sum_{x_{ij}} f_k(x_{ij})p(x_{ij}|u_i,v)p(v)dv.\end{aligned}$$

Since the query summation no longer depends on the j index, omit the average over the j query indices,

$$\begin{aligned}\mathbb{E}_{x,v}[\epsilon_{ik}] &= \frac{1}{m_i}\sum_{j|(i,j)\in T}\sum_{x_{ij}} f_k(x_{ij})p(x_{ij}|u_i,v_j) - \\ &\quad \int_v \sum_x f_k(x)p(x|u_i,v)p(v)dv. \qquad (8)\end{aligned}$$

After the simplifications illustrated above, the training sum (the first term in Equation (8)) is a function of the training item descriptors v_j for all j where $(i,j) \in T$. This function has Lipschitz constant $1/m_i$. Also, the second term in Equation (8) is the expectation of the function. Therefore, McDiarmid's inequality directly applies to this difference. The probability of $\mathbb{E}[\epsilon_{ik}]$ exceeding a constant t_2 is bounded by

$$p\left(\mathbb{E}[\epsilon_{ik}] \geq t_2\right) \leq \exp\left(-2m_i t_2^2\right). \qquad (9)$$

A union bound will be used to combine both deviations. Define the right-hand side of Equation (7) as

$$\frac{\delta}{2} = \exp\left(-\frac{2m_i\hat{m}_i t_1^2}{m_i + \hat{m}_i}\right).$$

Rewriting the above such that the corresponding deviation t_1 is a function of δ yields

$$t_1 = \sqrt{\frac{(m_i + \hat{m}_i)\ln\frac{2}{\delta}}{2m_i\hat{m}_i}}.$$

Similarly, let the right-hand side of the deviation bound in Equation (9) be $\delta/2$. The corresponding deviation as a function of δ is then

$$t_2 = \sqrt{\frac{\ln\frac{2}{\delta}}{2m_i}}.$$

Defining the total deviation as $\epsilon_{ik} = t_1 + t_2$ and applying a union bound completes the proof. □

B Maximum Entropy Dual

Since the number of queries can be much larger than the number of users and items, convert the maximum entropy problem into dual form via the Lagrangian

$$\begin{aligned}&\min_{\gamma,\lambda\in\mathbb{R}^+,\zeta\in\mathbb{R}}\max_p \sum_{ij\in Q} H(p_{ij}(x_{ij}))+ \\ &\sum_{\substack{ij\in Q\\ x_{ij}}} p_{ij}(x_{ij})\ln p_0(x_{ij}) - \sum_{ij\in Q}\zeta_{ij}(\sum_{x_i} p_{ij}(x_{ij}) - 1) \\ &+\sum_{k,i}(\gamma_{ik}^+ - \gamma_{ik}^-)\left(\frac{1}{\hat{m}_i}\sum_{j|(i,j)\in Q}\mathbb{E}_{x_{ij}}[f_k(x_{ij})] - \mu_{ik}\right) \\ &+\sum_{k,j}(\lambda_{jk}^+ - \lambda_{jk}^-)\left(\frac{1}{\hat{n}_j}\sum_{i|(i,j)\in Q}\mathbb{E}_{x_{ij}}[f_k(x_{ij})] - \nu_{jk}\right) \\ &+(\gamma_{ik}^+ + \gamma_{ik}^-)\alpha_i + (\lambda_{jk}^+ + \lambda_{jk}^-)\beta_j.\end{aligned}$$

Above, we define $\mathbb{E}_{x_{ij}}[f_k(x_{ij})] = \sum_{x_{ij}} p_{ij}(x_{ij})f_k(x_{ij})$ and use $p_{ij}(x_{ij})$ as shorthand for the conditional probability $p(x_{ij}|u_i,v_j)$. The Lagrange multipliers $\gamma_{ik}^{\pm}$ and $\lambda_{jk}^{\pm}$ correspond to the positive and negative absolute value constraints for the user and item averages. The ζ_{ij} multipliers correspond to equality constraints that force distributions to normalize.

The probabilities and the normalization multipliers can be solved for analytically resulting in the much simpler dual minimization program $\min_{\gamma,\lambda\geq 0}\mathcal{D}$ where the dual cost function is given by

$$\begin{aligned}\mathcal{D} &= \sum_{ik}(\gamma_{ik}^+ + \gamma_{ik}^-)\alpha_i - (\gamma_{ik}^+ - \gamma_{ik}^-)\mu_{ik} + \\ &\quad \sum_{kj}(\lambda_{jk}^+ + \lambda_{jk}^-)\beta_j - (\lambda_{jk}^+ - \lambda_{jk}^-)\nu_{jk} + \sum_{ij\in Q}\ln Z_{ij}.\end{aligned}$$

Here, Z_{ij} is the normalizing partition function for the estimated distribution for rating x_{ij}, and is defined as

$$Z_{ij} = \sum_{x_{ij}} p_0(x_{ij})\exp\left(\textstyle\sum_k\left(\frac{\gamma_{ik}^+ - \gamma_{ik}^-}{\hat{m}_i} + \frac{\lambda_{jk}^+ - \lambda_{jk}^-}{\hat{n}_j}\right)f_k(x_{ij})\right).$$

Once the Lagrange multipliers are found, the estimated probabilities are normalized Gibbs distributions of the form

$$p_{ij}(x_{ij}) \propto p_0(x_{ij})\exp\left(\textstyle\sum_k\left(\frac{\gamma_{ik}^+ - \gamma_{ik}^-}{\hat{m}_i} + \frac{\lambda_{jk}^+ - \lambda_{jk}^-}{\hat{n}_j}\right)f_k(x_{ij})\right).$$

Optimizing the cost function $\mathcal{D}$ requires taking partial derivatives which can be written in terms of normalized probabilities as follows

$$\begin{aligned}\frac{\partial\mathcal{D}}{\partial\gamma_{ik}^{\pm}} &= \alpha_i \mp \mu_{ik} \pm \frac{1}{\hat{m}_i}\sum_{j|(i,j)\in Q}\sum_{x_{ij}} p_{ij}(x_{ij})f_k(x_{ij}) \\ \frac{\partial\mathcal{D}}{\partial\lambda_{jk}^{\pm}} &= \beta_j \mp \nu_{jk} \pm \frac{1}{\hat{n}_j}\sum_{i|(i,j)\in Q}\sum_{x_{ij}} p_{ij}(x_{ij})f_k(x_{ij}).\end{aligned}$$

While previous ℓ_1-regularized Maxent methods have combined the positive and negative absolute value Lagrange multipliers (Dudík et al., 2007), we found that on our data, this led to numerical issues during optimization. Instead, we optimize both the positive and negative multipliers even though only one will be active at the solution. To reduce computation time, we use a simple cutting plane procedure. We initialize the problem at the prior distribution where all Lagrange multipliers are set to zero and find the worst violated constraints. We solve the dual, fixing all Lagrange multipliers at zero except the most violated constraints, and continue increasing the constraint set until all primal constraints are satisfied. In the worst case, this method eventually must solve a problem with half of the Lagrange multipliers active. Typically, we only need to optimize a much smaller subset. We solve the optimizations using the LBFGS-b optimizer (Zhu et al., 1997).