Large-Margin Matrix Factorization

Nathan Srebro **Jason Rennie** **Tommi Jaakkola**
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA, USA
nati,jrennie,tommi@mit.edu

Abstract

We present a novel approach to collaborative prediction, using low-norm instead of low-rank factorizations. The approach is inspired by, and has strong connections to, large-margin linear discrimination. We show how to learn low-norm factorizations by solving a semi-definite program, and present generalization error bounds based on analyzing the Rademacher complexity of low-norm factorizations.

1 Introduction

Fitting a target matrix Y with a low-rank matrix X by minimizing the sum-squared error is a common approach to modeling tabulated data, and can be done explicitly in terms of the singular value decomposition of Y. It is often desirable, though, to minimize a different loss function: loss corresponding to a specific probabilistic model (where X are the mean parameters, as in pLSA [1], or the natural parameters [2]); or loss functions such as hinge loss appropriate for binary or discrete ordinal data. Loss functions other than squared-error yield non-convex optimization problems with multiple local minima. Even with a squared-error loss, when only some of the entries in Y are observed, as is the case for collaborative filtering, local minima arise and SVD techniques are no longer applicable [3].

Low-rank approximations constrain the dimensionality of the factorization $X = UV'$. Other constraints, such as sparsity and non-negativity [4], have also been suggested for better capturing the structure in Y, and also lead to non-convex optimization problems.

In this paper we suggest regularizing the factorization by constraining the norm of U and V—constraints that arise naturally when matrix factorizations are viewed as feature learning for large-margin linear prediction (Section 2). Unlike low-rank factorizations, such constraints lead to *convex* optimization problems that can be formulated as semi-definite programs (Section 4). Throughout the paper, we focus on using low-norm factorizations for "collaborative prediction": predicting unobserved entries of a target matrix Y, based on a subset S of observed entries Y_S. In Section 5, we present generalization error bounds for collaborative prediction using low-norm factorizations.

2 Matrix Factorization as Feature Learning

Using a low-rank model for collaborative prediction [5, 6, 3] is straightforward: A low-rank matrix X is sought that minimizes a loss versus the observed entries Y_S. Unobserved

entries in Y are predicted according to X. Methods differ in how they relate real-valued entries in X to preferences in Y, and in the associated measure of discrepancy. For example, entries in X can be seen as parameters (either mean parameters [5, 6] or natural parameters [3, 7]) for a probabilistic model of the entries in Y, and a maximum likelihood criterion used. Alternatively, other loss functions, such as squared error [8, 3], can be minimized. Matrices of rank at most k are those that can be factored into $X = UV'$, $U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{m\times k}$, and so seeking a low-rank matrix is equivalent to seeking a low-dimensional factorization. The matrices U and V can be interpreted as "factors" or, depending on the measure of discrepancy, distributions of latent variables or mixture components.

If one of the matrices, say U, is fixed, and only the other matrix V' needs to be learned, then fitting each column of the target matrix Y is a separate linear prediction problem. Each row of U functions as a "feature vector", and each column of V' is a linear predictor, predicting the entries in the corresponding column of Y based on the "features" in U.

In collaborative prediction, both U and V are unknown and need to be estimated. This can be thought of as learning feature vectors (rows in U) for each of the rows of Y, enabling good linear prediction across all of the prediction problems (columns of Y) concurrently, each with a different linear predictor (columns of V'). The features are learned without any external information or constraints which is impossible for a single prediction task (we would use the labels as features). The underlying assumption that enables us to do this in a collaborative filtering situation is that the prediction tasks (columns of Y) are *related*, in that the same features can be used for all of them, though possibly in different ways.

The symmetric view, of learning features for the column enabling good linear prediction of the rows, is equally valid.

Low-rank collaborative prediction corresponds to regularizing by limiting the dimensionality of the feature space—each column is a linear prediction problem in a k-dimensional space. Instead, we suggest allowing an unbounded dimensionality for the feature space, and regularizing by requiring a low-norm factorization, while predicting with large-margin.

Consider adding to the loss a penalty term which is the sum of squares of entries in U and V, i.e. $\|U\|^2_{\text{Fro}} + \|V\|^2_{\text{Fro}}$ ($\|\|_{\text{Fro}}$ denotes the Frobenius norm). Each "conditional" problem (fitting U given V and vice versa) again decomposes into a collection of standard, this time regularized, linear prediction problems. With an appropriate loss function, or constraints on the observed entries, these correspond to large-margin linear discrimination problems. For example, if we learn a binary observation matrix by minimizing a hinge loss plus such a regularization term, each conditional problem decomposes into a collection of SVMs.

3 Large-Margin Matrix Factorizations

Matrices with a factorization $X = UV'$, where U and V have low Frobenius norm (recall that the dimensionality of U and V is no longer bounded!), can be characterized in several equivalent ways, and are known as low *nuclear norm* matrices:

Definition 1. *The nuclear norm $\|X\|_\Sigma$ is the sum of the singular values of X.*

Lemma 1. $\|X\|_\Sigma = \min_{X=UV'} \|U\|_{Fro} \|V\|_{Fro} = \min_{X=UV'} \frac{1}{2}(\|U\|^2_{Fro} + \|V\|^2_{Fro})$

The characterization in terms of the singular value decomposition allows us to characterize low nuclear norm matrices as the convex hull of low-norm rank-one matrices:

Lemma 2. $\{X \mid \|X\|_\Sigma \leq B\} = \operatorname{conv}\left\{uv' \mid u \in \mathbb{R}^n, v \in \mathbb{R}^m, |u|^2 = |v|^2 = B\right\}$

In particular, the nuclear norm is a convex function, and the set of bounded nuclear norm

matrices is a convex set. For convex loss functions, seeking a bounded nuclear norm matrix minimizing the loss versus some target matrix is a convex optimization problem.

This contrasts sharply with minimizing loss over low-rank matrices—a non-convex problem. Although the sum-squared error versus a *fully observed* target matrix can be minimized efficiently using the SVD (despite the optimization problem being non-convex!), minimizing other loss functions, or even minimizing a squared loss versus a partially observed matrix, is a difficult optimization problem with multiple local minima [3].

In fact, the nuclear norm has been suggested as a convex surrogate to the rank for various rank-minimization problems [9]. Here, we justify the nuclear norm directly, both as a natural extension of large-margin methods and by providing generalization error bounds.

To simplify presentation, we focus on binary labels, $Y \in \{\pm 1\}^{n \times m}$. We consider *hard-margin matrix factorization*, where we seek a minimum nuclear norm matrix X that matches the observed labels with a margin of one:

$$\begin{array}{l} \text{minimize } \|X\|_\Sigma \\ \text{subject to } Y_{ia} X_{ia} \geq 1 \text{ for all } ia \in S \end{array} \tag{1}$$

We also consider *soft-margin* learning, where we minimize a trade-off between the nuclear norm of X and its hinge-loss relative to Y_S:

$$\text{minimize } \|X\|_\Sigma + c \sum_{ia \in S} \max(0, 1 - Y_{ia} X_{ia}). \tag{2}$$

As in large-margin linear discrimination, there is an inverse dependence between the norm and the margin. Fixing the margin and minimizing the nuclear norm is equivalent to fixing the nuclear norm and maximizing the margin. As in large-margin discrimination with certain infinite dimensional (e.g. radial) kernels, the data is always separable with sufficiently high nuclear norm (a nuclear norm of $\sqrt{n|S|}$ is sufficient to attain a margin of one).

4 Learning Large-Margin Matrix Factorizations

In this section we investigate the optimization problem of learning with low nuclear norm matrices, focusing on learning a binary target matrix, and see how this optimization problem can be written as a semi-definite program.

Bounding the nuclear norm of UV' by $\frac{1}{2}(\|U\|_{\text{Fro}}^2 + \|V\|_{\text{Fro}}^2)$, we can characterize the nuclear norm in terms of the trace of a positive semi-definite matrix:

Lemma 3 ([9, Lemma 1]). *For any $X \in \mathbb{R}^{n \times m}$ and $t \in \mathbb{R}$: $\|X\|_\Sigma \leq t$ iff there exists $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{m \times m}$ such that* [1] $\left[\begin{smallmatrix} A & X \\ X' & B \end{smallmatrix}\right] \succcurlyeq 0$ *and* $\operatorname{tr} A + \operatorname{tr} B \leq 2t$.

Proof. Note that for any matrix W, $\|W\|_{\text{Fro}} = \operatorname{tr} WW'$. If $\left[\begin{smallmatrix} A & X \\ X' & B \end{smallmatrix}\right]$ is p.s.d. with $\operatorname{tr} A + \operatorname{tr} B \leq 2t$, we can write it as a product $\left[\begin{smallmatrix} U \\ V \end{smallmatrix}\right]\left[\begin{smallmatrix} U' & V' \end{smallmatrix}\right]$. We have $X = UV'$ and $\frac{1}{2}(\|U\|_{\text{Fro}}^2 + \|V\|_{\text{Fro}}^2) = \frac{1}{2}(\operatorname{tr} A + \operatorname{tr} B) \leq t$, establishing $nnorm X \leq t$. Conversely, if $\|X\|_\Sigma \leq t$ we can write it as $X = UV'$ with $\operatorname{tr} UU' + \operatorname{tr} VV' \leq 2t$ and consider the p.s.d. matrix $\left[\begin{smallmatrix} UU' & X \\ X' & VV' \end{smallmatrix}\right]$. □

Lemma 3 can be used in order to formulate minimizing the nuclear norm as a semi-definite optimization problem (SDP). The hard-margin matrix factorization problem (1) can be

[1] $A \succcurlyeq 0$ denotes A is positive semi-definite

written as:

$$\min \frac{1}{2}(\operatorname{tr} A + \operatorname{tr} B) \text{ s.t. } \begin{bmatrix} A & X \\ X' & B \end{bmatrix} \succcurlyeq 0 \quad y_{ia} X_{ia} \geq 1 \quad \forall ia \in S \tag{3}$$

And introducing slack, soft-margin matrix factorization (2), can be written as:

$$\min \frac{1}{2}(\operatorname{tr} A + \operatorname{tr} B) + c \sum_{ia \in S} \xi_{ia} \text{ s.t. } \begin{bmatrix} A & X \\ X' & B \end{bmatrix} \succcurlyeq 0, \quad \begin{matrix} y_{ia} X_{ia} \geq 1 - \xi_{ia} \\ \xi_{ia} \geq 0 \end{matrix} \ \forall ia \in S \tag{4}$$

Associating a dual variable Q_{ia} with each constraint on X_{ia}, the dual of (4) is [10]:

$$\max \sum_{ia \in S} Q_{ia} \text{ s.t. } \begin{bmatrix} I & (-Q \otimes Y) \\ (-Q \otimes Y)' & I \end{bmatrix} \succcurlyeq 0, \quad 0 \leq Q_{ia} \leq c \tag{5}$$

where $Q \otimes Y$ denotes the sparse matrix $(Q \otimes Y)_{ia} = Q_{ia} Y_{ia}$ for $ia \in S$ and zeros elsewhere. The dual of the hard-margin problem is similar, but without the box constraints $q_{ia} \leq c$. In either case, problem is strictly feasible, and there is no duality gap.

The p.s.d. constraint in the dual (5) is equivalent to bounding the spectral norm of $Q \otimes Y$, and the dual can also be written as an optimization problem subject to a bound on the spectral norm, i.e. a bound on the singular values of $Q \otimes Y$:

$$\max \sum_{ia \in S} Q_{ia} \text{ s.t. } \begin{matrix} \|Q \otimes Y\|_2 \leq 1 \\ 0 \leq Q_{ia} \leq c \quad \forall ia \in S \end{matrix} \tag{6}$$

In typical collaborative prediction problems, we observe only a small fraction of the entries in a large target matrix. Such a situation translates to a sparse dual semi-definite program, with the number of variables equal to the number of observed entries. Large-scale SDP solvers can take advantage of such sparsity.

4.1 Using the dual solution

Most SDP solvers use internal point methods and return a pair of primal and dual optimal solutions. The prediction matrix X^* minimizing (2) is part of the primal optimal solution of (4), and can be extracted from it directly.

Nevertheless, it is interesting to study how the optimal prediction matrix X^* can be directly recovered from a dual optimal solution Q^* alone. Although unnecessary when relying on standard internal point SDP solvers, this might enable us to use specialized optimization methods, taking advantage of the simple structure of the dual.

Recovering X^* from Q^* As for linear programming, recovering a primal optimal solution directly from a dual optimal solution is not always possible for SDPs in general. However, at least for the hard-margin problem (1) this is possible, and we describe below how an optimal prediction matrix X^* can be recovered from a dual optimal solution Q^* by calculating a singular value decomposition and solving linear equations.

Given a dual optimal Q^*, consider its singular value decomposition $Q^* \otimes Y = U \Lambda V'$. Recall that all singular values of $Q^* \otimes Y$ are bounded by one, and consider only the columns $\tilde{U} \in \mathbb{R}^{n \times p}$ of U and $\tilde{V} \in \mathbb{R}^{m \times p}$ of V with singular values one. It is possible to show [10], using complimentary slackness, that for some matrix $R \in \mathbb{R}^{p \times p}$, $X^* = \tilde{U} R R' \tilde{V}'$ is an optimal solution to the maximum margin matrix factorization problem (2). Furthermore, $\frac{p(p+1)}{2}$ is bounded above by the number of non-zero Q^*_{ia}. When $Q^*_{ia} > 0$, and assuming

hard-margin constraints, i.e. no box constraints in the dual, complimentary slackness dictates that $X^*_{ia} = \tilde{U}_i RR' \tilde{V}'_a = Y_{ia}$, providing us with a linear equation on the $\frac{p(p+1)}{2}$ entries in the symmetric RR'. For hard-margin matrix factorization, we can therefore recover the entries of RR' by solving a system of linear equations, with a number of variables bounded by the number of observed entries.

Recovering specific entries The approach described above requires solving a large system of linear equations (with as many variables as observations). Furthermore, especially when the observations are very sparse (only a small fraction of the entries in the target matrix are observed), the dual solution is much more compact then the prediction matrix: the dual involves a single number for each *observed* entry. It might be desirable to avoid storing the prediction matrix X^* explicitly, and calculate a desired entry $X^*_{i_0 a_0}$, or at least its sign, directly from the dual optimal solution Q^*.

Consider adding the constraint $X_{i_0 a_0} > 0$ to the primal SDP (4). If there exists an optimal solution X^* to the original SDP with $X^*_{i_0 a_0} > 0$, then this is also an optimal solution to the modified SDP, with the same objective value. Otherwise, the optimal solution of the modified SDP is not optimal for the original SDP, and the optimal value of the modified SDP is higher (worse) then the optimal value of the original SDP.

Introducing the constraint $X_{i_0 a_0} > 0$ to the primal SDP (4) corresponds to introducing a new variable $Q_{i_0 a_0}$ to the dual SDP (5), appearing in $Q \otimes Y$ (with $Y_{i_0 a_0} = 1$) but *not* in the objective. In this correspondingly modified dual, the optimal solution Q^* of the original dual would always be feasible. But, if $X^*_{i_0 a_0} \leq 0$ in all primal optimal solutions, and the modified primal SDP has a higher value, then so does the dual, and Q^* is no longer optimal for the new dual. By checking the optimality of Q^* for the modified dual, e.g. by attempting to re-optimize it, we can recover the sign of $X^*_{i_0 a_0}$.

We can repeat this test once with $Y_{i_0 a_0} = 1$ and once with $Y_{i_0 a_0} = -1$, corresponding to $X_{i_0 a_0} < 0$. If $Y_{i_0 a_0} X^*_{i_0 a_0} < 0$ (in all optimal solutions), then the dual solution can be improved by introducing $Q_{i_0 a_0}$ with a sign of $Y_{i_0 a_0}$.

4.2 Predictions for new users

So far, we have assumed that learning is done on the known entries in all rows. It is commonly desirable to predict entries in a new partially observed row of Y (a new user in a collaborative filtering task), not included in the original training set. This essentially requires solving a "conditional" problem, where V is already known, and a new row of U is learned (the predictor for the new user) based on a new partially observed row of X. Using large-margin matrix factorization, this is a standard SVM problem.

5 Generalization Error Bounds for Low Norm Matrix Factorizations

Similarly to standard feature-based prediction approaches, collaborative prediction methods can also be analyzed in terms of their generalization ability: How confidently can we predict all the entries of Y based on our error on the observed entries Y_S? We present here generalization error bounds that hold for *any* target matrix Y, and for a random subset of observations S, and bound the average error across all entries by the observed empirical error. The central assumption, paralleling the i.i.d. source assumption for standard feature-based prediction, is that the observed subset S is picked uniformly at random.

Theorem 4. *For all target matrices $Y \in \{\pm 1\}^{n \times m}$ and sample sizes $|S| > n \log n$, and for a uniformly selected sample S of $|S|$ entries in Y, with probability at least $1 - \delta$ over*

the sample selection, the following holds for all matrices $X \in \mathbb{R}^{n\times m}$:

$$\frac{1}{nm}\sum \textit{loss}(X_{ia};Y_{ia}) < \frac{1}{|S|}\sum_{ia\in S}\textit{loss}(X_{ia};Y_{ia}) + KL\frac{\|X\|_\Sigma}{\sqrt{nm}}\sqrt[4]{\ln m}\sqrt{\frac{(n+m)\ln n}{|S|}} + \sqrt{\frac{\ln(1+|\log\|X\|_\Sigma|)}{|S|}} + \sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

Where K is a universal constant that does not depend on Y,n,m, the loss function or any other quantity, and loss is Lipschitz continuous with constant L, and we assume $n \geq m$.

By bounding the zero-one error in terms of a picewise linear margin loss $\text{loss}(x,y) = \max(0,\min(yx-1,1))$, which in turn is bounded by the zero-one margin loss, the generalization error bound can be specialized to bounding the true zero-one error in terms of the empirical zero-one margin error:

Corollary 5. *For all target matrices $Y \in \{\pm 1\}^{n\times m}$ and sample sizes $|S| > n\log n$, and for a uniformly selected sample S of $|S|$ entries in Y, with probability at least $1-\delta$ over the sample selection, the following holds for all matrices $X \in \mathbb{R}^{n\times m}$ and all $\gamma > 0$:*

$$\frac{1}{nm}|\{ia|X_{ia}Y_{ia}\leq 0\}| < \frac{1}{|S|}|\{ia\in S|X_{ia}Y_{ia}\leq\gamma\}| + K\frac{\|X\|_\Sigma}{\gamma\sqrt{nm}}\sqrt[4]{\ln m}\sqrt{\frac{(n+m)\ln n}{|S|}} + \sqrt{\frac{\ln(1+|\log\|X\|_\Sigma/\gamma|)}{|S|}} + \sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

To understand the scaling of this bound, it is useful to consider the scaling of the nuclear norm for matrices that can be factored into $X = UV'$ where the norm of each row of U and V is bounded by r. The nuclear norm of such matrices is at most $r^2\sqrt{nm}$, leading to a complexity term of r^2. Recall that the conditional problem, where V is fixed and only U is learned, is a collection of low-norm (large-margin) linear prediction problems. When the norms of rows in U and V are bounded by r, a similar generalization error bound on the conditional problem would include the term $r^2\sqrt{\frac{n}{|S|}}$, matching the term in Theorem 4 up to log-factors. We see, then, that learning *both* U and V does not introduce significantly more structural risk than learning just one of them.

Also of interest are low-rank matrices, for which $\|X\|_\Sigma \leq \sqrt{\text{rank}\,X}\,\|X\|_{\text{Fro}}$. In particular, for rank-k X with entries bounded by B, we have (for fixed B and k):

$$\frac{1}{nm}\sum \text{loss}(X_{ia};Y_{ia}) < \frac{1}{|S|}\sum_{ia\in S}\text{loss}(X_{ia};Y_{ia}) + KLB\sqrt[4]{\ln m}\sqrt{\frac{k(n+m)\ln n+\ln(4/\delta)}{|S|}}$$

This is the best (up to log factors) that can be achieved without explicitly bounding the loss function. But for bounded loss functions, analyzing the covering number of bounded low-rank matrices directly, yields a bound that scales only logarithmically with B.

6 Implementation and Experiments

Ratings In many collaborative prediction tasks, the labels are not binary, but rather are discrete "ratings" in several ordered levels (e.g. one star through five stars). Separating R levels by thresholds $-\infty = \theta_0 < \theta_1 < \cdots < \theta_R = \infty$, and generalizing hard-margin constraints for binary labels, one can require $\theta_{Y_{ia}} + 1 \leq X_{ia} \leq \theta_{Y_{ia}+1} - 1$. A soft-margin version of these constraints, with slack variables for the two constraints on each observed rating, corresponds to a generalization of the hinge loss which is a convex bound

on the zero/one level-agreement error [11]. To obtain a loss which is a convex bound on the mean-absolute-error (the difference, in levels, between the predicted level and the true level), we introduce $R-1$ slack variables for each observed rating—one for each of the $R-1$ constraints $X_{ia} \geq \theta_r$ for $r < Y_{ia}$ and $X_{ia} \leq \theta_r$ for $r \geq Y_{ia}$. Both of these soft-margin problems ("immediate-threshold" and "all-threshold") can be formulated as SDPs similar to (4)-(5). Furthermore, it is straightforward to learn also the thresholds (they appear as variables in the primal, and correspond to constraints in the dual)—either a single set of thresholds for the entire matrix, or a separate threshold vector for each row of the matrix (each "user"). Doing the latter allows users to "use ratings differently" and alleviates the need to normalize the data.

Experiments We conducted preliminary experiments on a subset of the MovieLens dataset[2], consisting of the 100 users and 100 movies with the most ratings. The ratings are on a discrete scale of one through five, and we experimented with both generalizations of the hinge loss described above, allowing per-user thresholds. We used CSDP [12] to solve the resulting SDPs[3]. We compared against methods described in [13], randomly selecting 50% of the entries for training and 50% for testing. We tested a range of regularization parameters (C/K) and present the best zero-one agreement error (ZOE) and mean-absolute-error (MAE) result for each method, with the regularization parameters attaining it.

	all-θ LMMF,c=0.2	immediate-θ LMMF,c=0.3	K-medians K=2	WLRA K=1	WLRA K=2
MAE	0.508 / **0.670**	0.621 / 0.715	0.620 / 0.674	0.679 / 0.698	0.622 / 0.714
ZOE	0.450 / 0.553	0.462 / **0.542**	0.510 / 0.558	0.550 / 0.559	0.519 / 0.553

Table 1: Lowest train/test errors for various methods.

7 Discussion

Learning large-margin matrix factorizations requires solving a sparse semi-definite program. We experimented with generic SDP solvers, and were able to learn with up to tens of thousands of labels. We propose that just as generic QP solvers do not perform well on SVM problems, special purpose techniques, taking advantage of the very simple structure of the dual (5), might be necessary in order to solve large-scale large-margin matrix factorization problems.

SDPs were recently suggested for a related, but different, problem: learning the features (or equivalently, kernel) that are best for a *single* prediction task [14]. This task is hopeless if the features are completely unconstrained, as they are in our formulation. Gert *et al* suggest constraining the allowed features, e.g. to a linear combination of a few "base feature spaces" (or base kernels), which represent the external information necessary to solve a single prediction problem. It is possible to combine the two approaches, seeking constrained features for multiple related prediction problems, as a way of combining external information (e.g. details of users and of items) and collaborative information.

An alternate method for introducing external information into our formulation is by adding to U and/or V additional fixed (non-learned) columns representing the external features. This method degenerates to standard SVM learning when Y is a vector rather than a matrix.

A variant of the approach suggested here (which can also be written as a SDP) is bounding the *maximum* norm of rows in the factorization, replacing the nuclear norm with

[2] `http://www.cs.umn.edu/Research/GroupLens/`

[3] Solving with immediate-threshold loss took about 25 CPU minutes on a 3.06GHz Intel Xeon. Solving with all-threshold loss took up to eight hours. The MATLAB code used to generate the SDPs (in SDPA format) is available at `www.ai.mit.edu/~nati/lmmf`

$\|X\|_{\max} = \min_{X=UV'}(\max_i |U_i|)(\max_a |V_a|)$ where U_i, V_a are rows of U, V. Low-max-norm discrimination has a clean geometric interpretation. First, note that predicting the target matrix with the signs of a rank-k matrix corresponds to mapping the "items" (columns) to points in $\mathbb{R}^k$, and the "users" (rows) to homogeneous hyperplanes, such that each user's hyperplane separates his positive items from his negative items. Hard-margin low-max-norm prediction corresponds to mapping the users and items to points and hyperplanes in a high-dimensional unit sphere such that each user's hyperplane separates his positive and negative items with a large-margin (the margin being the inverse of the max-norm).

An important limitation of the approach we have described, is that observed entries are assumed to be uniformly sampled. This is made explicit in the generalization error bounds. Such an assumption is typically unrealistic, as, e.g., users tend to rate items they like. At an extreme, it is often desirable to make predictions based only on positive samples. Even in such situations, it is still possible to learn a low-norm factorization, by using appropriate loss functions, e.g. derived from probabilistic models incorporating the observation process. However, obtaining generalization error bounds in this case is much harder. Simply allowing an arbitrary sampling distribution and calculating the expected loss based on this distribution (which is not possible with the nuclear norm, but might be possible with the max-norm) is not satisfying, as this would guarantee low error on items the user is likely to want anyway, but not on items we predict he would like.

A Proof of Theorem 4

We sketch here the main arguments of the proof of Theorem 4. Complete details are available in [15]. To prove the theorem, we consider matrices $X \in \mathbb{R}^{n\times m}$ as functions $X : [n] \times [m] \to \mathbb{R}$ from index pairs to entries in the matrix, and bound their Rademacher complexity [16] as such. The proof is then an application of Theorem 2 of [16].

In order to calculate the Rademacher complexity of matrices with bounded nuclear norm, we calculate the Rademacher complexity of unit-norm rank-one matrices, $\mathcal{X}_1[1] \doteq \{uv' \mid u \in \mathbb{R}^n, v \in \mathbb{R}^m, |u| = |v| = 1\}$, and use the fact that the Rademacher complexity does not change when we take the convex hull of this class. We first analyze the empirical Rademacher complexity, for any fixed sample S, possibly with repeating index pairs. We then bound the (average) Rademacher complexity for a sample of $|S|$ index pairs drawn uniformly at random from $[n] \times [m]$ (with repetitions). The resulting generalization error bound applies to samples selected by this process, and therefore also bounds the more concentrated situation of samples drawn without repetitions.

The Empirical Rademacher Complexity For a sample $S = \{(i_1, a_1), (i_2, a_2), \ldots\}$ of $|S|$ index pairs, we need to take an expectation over random signs σ_α for each appearance of an index pair (i_α, a_α) in the sample. For each index pair (i, a) we will denote s_{ia} the number of times it appears in S (possibly zero), and consider instead the random variable $\sigma_{ia} = \sum_{(i_\alpha,a_\alpha)=(i,a)} \sigma_\alpha$, the sum of s_{ia} random signs. We can now calculate:

$$\hat{R}_S(\mathcal{X}_1[1]) = \mathbf{E}_\sigma \left[\sup_{|u|=|v|=1} \left| \frac{2}{|S|} \sum_{i,a} \sigma_{ia} u_i v_a \right| \right] = \frac{2}{|S|} \mathbf{E}_\sigma \left[\sup_{|u|=|v|=1} |u'\sigma v| \right] = \frac{2\mathbf{E}_\sigma \left[\|\sigma\|_2 \right]}{|S|}$$

where σ is an $n \times m$ matrix of σ_{ia}. Using the Frobenius norm to bound the spectral norm, $\hat{R}_S(\mathcal{X}_1[1]) \leq \frac{2}{|S|}\mathbf{E}_\sigma \left[\|\sigma\|_{\text{Fro}} \right] \leq \frac{2}{|S|}\sqrt{\sum_{ia} \mathbf{E}_\sigma \left[\sigma_{ia}^2\right]} = \frac{2}{\sqrt{|S|}}$. As a supremum over all sample sets S, this bound is tight (and not very useful), but for uniformly distributed samples the expected Rademacher complexity is much lower.

Bounding $\mathbf{E}_\sigma\left[\|\sigma\|_2\right]$ Instead of using the Frobenius norm, we bound the expected spectral norm directly. We do so by applying Theorem 3.1 of [17], which bounds the expected spectral norm of matrices with entries of fixed magnitudes but random signs in terms of the maximum row and column magnitude norms. If S contains no repeated index pairs ($s_{ia} = 0$ or 1), we are already in this situation, as the magnitudes of σ are equal to s. When some index pairs are repeated, we consider a different random matrix, $\tilde{\sigma}$ which consists of sign flips of s_{ia}: $\tilde{\sigma}_{ia} = \epsilon_{ia} s_{ia}$ where ϵ_{ia} are i.i.d. unbiased signs. Using $\tilde{\sigma}$ instead of σ gives us an upper bound on the empirical Rademacher complexity [15]:

$$\hat{R}_S(\mathcal{X}_1[1]) \leq \frac{2}{|S|}\mathbf{E}_\epsilon\left[\|\tilde{\sigma}\|_2\right] \leq K(\ln m)^{\frac{1}{4}}\left(\max_i |s_{i\cdot}| + \max_a |s_{\cdot a}|\right)$$

where $s_{i\cdot}$ and $s_{\cdot a}$ are row and column vectors of the matrix s, and K is the absolute constant guaranteed by Theorem 3.1 of [17].

Bounding the Row and Column Norms of a Uniformly Random Sample For the worst samples, the norm of a single row or column vector of s might be as high as $|S|$, but for random uniformly drawn samples, we would expect the norm of row vectors to be roughly $\frac{|S|}{n}$ and of column vectors to be roughly $\frac{|S|}{m}$. To make this estimate precise we proceed in two steps. We first use Bernstein's inequality to bound the maximum value of s_{ia}, uniformly over all index pairs: $\Pr_S\left(\max_{ia} s_{ia} > 9\ln n\right) \leq \frac{1}{|S|}$. When the maximum entry in s is bounded, the norm of a row can be bounded by the number of observations in the row. In the second step we use Bernstein's inequality again to bound the expected maximum number of observations in a row (similarly column) by $6(\frac{|S|}{n} + \ln|S|)$. Combining these results we can bound the Rademacher complexity, for a random sample set where each index pair is chosen uniformly and independently at random:

$$\begin{aligned} R_{|S|}(\mathcal{X}_1[1]) = \mathbf{E}_S\left[R_S\right] &\leq \Pr\left(\max_{ia} s_{ia} > 9\ln n\right)\sup_S R_S + \mathbf{E}_S\left[R_S \,\middle|\, \max_{ia} s_{ia} \leq 9\ln n\right] \\ &\leq \frac{2}{|S|} + \frac{K}{|S|}(\ln m)^{\frac{1}{4}}\left(\sqrt{9\ln n 6(\frac{|S|}{n} + \ln|S|)} + \sqrt{9\ln n 6(\frac{|S|}{m} + \ln|S|)}\right) \end{aligned}$$

Taking the convex hull of $\mathcal{X}_1[1]$, scaling by M and rearranging terms:

Theorem 6. *For some universal constant K, the Rademacher complexity of matrices of nuclear norm at most M, over uniform samplings of index pairs, is at most (for $n \geq m$):*

$$R(\mathcal{X}[M]) \leq K\frac{M}{\sqrt{nm}}(\ln m)^{\frac{1}{4}}\sqrt{\frac{(n+m+\frac{nm}{|S|}\ln n)\ln n}{|S|}}$$

When $|S| > n\ln n$, the last term can be subsumed in the constant K.

Acknowledgments We would like to thank Sam Roweis for pointing out [9] and Dmitry Panchenko for guiding us through Rademacher complexities.

References

[1] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal*, 42(1):177–196, 2001.

[2] M. Collins, S. Dasgupta, and R. Schapire. A generalization of principal component analysis to the exponential family. In *NIPS*, 2001.

[3] Nathan Srebro and Tommi Jaakkola. Weighted low rank approximation. In *20th International Conference on Machine Learning*, 2003.

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

[5] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, 2004.

[6] Benjamin Marlin. Modeling user rating profiles for collaborative filtering. In *NIPS*17*, 2004.

[7] Benjamin Marlin and Richard S. Zemel. The multiple multiplicative factor model for collaborative filtering. In *To appear in ICML*, 2004.

[8] Yossi Azar, Amos Fiat, Anna R. Karlin, Frank McSherry, and Jared Saia. Spectral analysis of data. In *ACM Symposium on Theory of Computing*, pages 619–626, 2001.

[9] Maryam Fazel, Haitham Hindi, and Stephen P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings American Control Conference*, volume 6, 2001.

[10] Finding large-margin matrix factorizations using semi definite programming. `www.ai.mit.edu/~nati/lmmf`.

[11] Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. In *NIPS*14*, 2003.

[12] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

[13] B. Marlin. Collaborative filtering: A machine learning perspective. Master's thesis, University of Toronto, 2004.

[14] Gert R.G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[15] Generalizatoin error bounds for large-margin matrix factorizations. `www.ai.mit.edu/~nati/lmmf`.

[16] Dmitry Panchenko and Vladimir Koltchinskii. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30(1), 2002.

[17] Yoav Seginer. The expected norm of random matrices. *Comb. Probab. Comput.*, 9(2):149–166, 2000.