Collaborative Filtering in a Non-Uniform World: Learning with the Weighted Trace Norm

Ruslan Salakhutdinov
Brain and Cognitive Sciences and CSAIL, MIT
Cambridge, MA 02139
rsalakhu@mit.edu

Nathan Srebro
Toyota Technological Institute at Chicago
Chicago, Illinois 60637
nati@ttic.edu

Abstract

We show that matrix completion with trace-norm regularization can be significantly hurt when entries of the matrix are sampled non-uniformly, but that a properly weighted version of the trace-norm regularizer works well with non-uniform sampling. We show that the weighted trace-norm regularization indeed yields significant gains on the highly non-uniformly sampled Netflix dataset.

1 Introduction

Trace-norm regularization is a popular approach for matrix completion and collaborative filtering, motivated both as a convex surrogate to the rank [7, 6] and in terms of a regularized infinite factor model with connections to large-margin norm-regularized learning [17, 1, 15].

Current theoretical guarantees on using the trace-norm for matrix completion assume a uniform sampling distribution over entries of the matrix [18, 6, 5, 13]. In a collaborative filtering setting, where rows of the matrix represent e.g. users and columns represent e.g. movies, this corresponds to assuming all users are equally likely to rate movies and all movies are equally likely to be rated. This of course cannot be further from the truth, as invariably some users are more active than others and some movies are rated by many people while others are rarely rated.

In this paper we show, both analytically and through simulations, that this is not a deficiency of the proof techniques used to establish the above guarantees. Indeed, a non-uniform sampling distribution can lead to a significant deterioration in prediction quality and an increase in the sample complexity. Under non-uniform sampling, as many as $\Omega(n^{4/3})$ samples might be needed for learning even a simple (e.g. orthogonal low rank) $n \times n$ matrix. This is in sharp contrast to the uniform sampling case, in which $\tilde{O}(n)$ samples are enough. It is important to note that if the rank could be minimized directly, which is in general not computationally tractable, $\tilde{O}(n)$ samples would be enough to learn a low-rank model even under an arbitrary non-uniform distribution.

Our analysis further suggests a weighted correction to the trace-norm regularizer, that takes into account the sampling distribution. Although appearing at first as counter-intuitive, and indeed being the opposite of a previously suggested weighting [21], this weighting is well-motivated by our analytic analysis and we discuss how it corrects the problems that the unweighted trace-norm has with non-uniform sampling. We show how the weighted trace-norm indeed yields a significant improvement on the highly non-uniformly sampled Netflix dataset.

The only other work we are aware of that studies matrix completion under non-uniform sampling is work on *exact* completion (i.e. when the matrix is assumed to be *exactly* low rank) under power-law sampling [12]. Other then being limited to one specific distribution, the requirement of the matrix being exactly low rank is central to this work, and the results cannot be directly applied in the presence of even small noise. Empirically, the approach leads to deterioration in predictive performance on the Netflix data [12].

2 Complexity Control in terms of Matrix Factorizations

Consider the problem of predicting the entries of some unknown target matrix $Y \in \mathbb{R}^{n\times m}$ based on a random subset S of observed entries Y_S. For example, n and m may represent the number of users and the number of movies, and Y may represent a matrix of partially observed rating values. Predicting elements of Y can be done by finding a matrix X minimizing the training error, here measured as a squared error, and some measure $c(X)$ of complexity. That is, minimizing either:

$$\min_X \|X_S - Y_S\|_{\mathrm{F}}^2 + \lambda c(X) \tag{1}$$

or:

$$\min_{c(X)\le C} \|X_S - Y_S\|_{\mathrm{F}}^2 , \tag{2}$$

where Y_S, and similarly X_S, denotes the matrix "masked" by S, where $(Y_S)_{i,j} = Y_{i,j}$ if $(i,j) \in S$ and 0 otherwise. For now we ignore possible repeated entries in S and we also assume that $n \le m$ without loss of generality. The two formulations (1) and (2) are equivalent up to some (unknown) correspondence between λ and C, and we will be referring to them interchangeably.

A basic measure of complexity is the rank of X, corresponding to the minimal dimensionality k such that $X = U^\top V$ for some $U \in \mathbb{R}^{k\times n}$ and $V \in \mathbb{R}^{k\times m}$. Directly constraining the rank of X forms one of the most popular approaches to collaborative filtering. However, the rank is non-convex and hard to minimize. It is also not clear if a strict dimensionality constraint is most appropriate for measuring the complexity.

Trace-norm Regularization

Lately, methods regularizing the *norm* of the factorization $U^\top V$, rather than its dimensionality, have been advocated and were shown to enjoy considerable empirical success [14, 15]. This corresponds to measuring complexity in terms of the *trace-norm* of X, which can be defined equivalently either as the sum of the singular values of X, or as [7]:

$$\|X\|_{\mathrm{tr}} = \min_{X=U'V} \frac{1}{2}(\|U\|_{\mathrm{F}}^2 + \|V\|_{\mathrm{F}}^2), \tag{3}$$

where the dimensionality of U and V is not constrained. Beyond the modeling appeal of norm-based, rather than dimension-based, regularization, the trace-norm is a convex function of X and so can be minimized by either local search or more sophisticated convex optimization techniques.

Scaling

The rank, as a measure of complexity, does not scale with the size of the matrix. That is, even very large matrices can have low rank. Viewing the rank as a complexity measure corresponding to the number of underlying factors, if data is explained by e.g. two factors, then no matter how many rows ("users") and columns ("movies") we consider, the data will still have rank two. The trace-norm, however, does scale with the size of the matrix. To see this, note that the trace-norm is the ℓ_1 norm of the spectrum, while the Frobenius norm is the ℓ_2 norm of the spectrum, yielding:

$$\|X\|_{\mathrm{F}} \le \|X\|_{\mathrm{tr}} \le \|X\|_{\mathrm{F}} \sqrt{\mathit{rank}(X)} \le \sqrt{n}\,\|X\|_{\mathrm{F}} . \tag{4}$$

The Frobenius norm certainly increases with the size of the matrix, since the magnitude of each element does not decrease when we have more elements, and so the trace-norm will also increase. The above suggests measuring the trace-norm relative to the Frobenius norm. Without loss of generality, consider each target entry to be of roughly unit magnitude, and so in order to fit Y each entry of X must also be of roughly unit magnitude. This suggests scaling the trace-norm by $\sqrt{nm}$. More specifically, we study the trace-norm through the complexity measure:

$$\mathit{tc}(X) = \frac{\|X\|_{\mathrm{tr}}^2}{nm}, \tag{5}$$

which puts the trace-norm on a comparable scale to the rank. In particular, when each entry of X is, on-average, of unit magnitude (i.e. has unit variance) we have $1 \le \mathit{tc}(X) \le \mathit{rank}(X)$.

The relationship between $\mathit{tc}(X)$ and the rank is tight for "orthogonal" low-rank matrices, i.e. low-rank matrices $X = U^\top V$ where the rows of U and also the rows of V are orthogonal and of equal magnitudes. In order for the entries in Y to have unit magnitude, i.e. $\|Y\|_{\mathrm{F}}^2 = nm$, we have that rows

in U have norm $\sqrt{n/\sqrt{k}}$ and rows in V have norm $\sqrt{m/\sqrt{k}}$, yielding precisely $tc(X) = rank(X)$. Such an orthogonal low-rank matrix can be obtained, e.g., when entries of U and V are zero-mean i.i.d. Gaussian with variance $1/\sqrt{k}$, corresponding to unit-variance entries in X.

Generalization Guarantees

Another place where we can see that $tc(X)$ plays a similar role to $rank(X)$ is in the generalization and sample complexity guarantees that can be obtained for low-rank and low-trace-norm learning. If there is a low-rank matrix X^* achieving low average error relative to Y (e.g. if $Y = X^* + \text{noise}$), then by minimizing the training error subject to a rank constraint (a computationally intractable task), $|S| = \tilde{O}(rank(X^*)(n+m))$ samples are enough in order to guarantee learning a matrix X whose overall average error is close to that of X^* [16]. Similarly, if there is a low-trace-norm matrix X^* achieving low average error, then minimizing the training error and the trace-norm (a convex optimization problem), $|S| = \tilde{O}(tc(X^*)(n+m))$ samples are enough in order to guarantee learning a matrix X whose overall average error is close to that of X^* [18]. In these bounds $tc(X)$ plays precisely the same role as the rank, up to logarithmic factors.

In order to get some intuitive understanding of low-rank learning guarantees, it is enough to consider the number of parameters in the rank-k factorization $X = U^\top V$. It is easy to see that the number of parameters in the factorization is roughly $k(m+n)$ (perhaps a bit less due to rotational invariants). We therefore would expect to be able to learn X when we have roughly this many samples, as is indeed confirmed by the rigorous sample complexity bounds.

For low-trace-norm learning, consider a sample S of size $|S| \leq Cn$, for some constant C. Taking entries of Y to be of unit magnitude, we have $\|Y_S\|_{\text{F}} = \sqrt{|S|} \leq \sqrt{Cn}$ (recall that Y_S is defined to be zero outside S). From (4) we therefore have: $\|Y_S\|_{\text{tr}} \leq \sqrt{Cn} \cdot \sqrt{n} = \sqrt{C}n$ and so $tc(Y_S) \leq C$. That is, we can "shatter" any sample of size $|S| \leq Cn$ with $tc(X) = C$: no matter what the underlying matrix Y is, we can always perfectly fit the training data with a low trace-norm matrix X s.t. $tc(X) \leq C$, without generalizing at all outside S. On the other hand, we must allow matrices with $tc(X) = tc(X^*)$, otherwise we can not hope to find X^*, and so we can only constrain $tc(X) \leq C = tc(X^*)$. We therefore cannot expect to learn with less than $ntc(X^*)$ samples. It turns out that this is essentially the largest random sample that can be shattered with $tc(X) \leq C = tc(X^*)$. If we have more than this many samples we can start learning.

3 Trace-Norm Under a Non-Uniform Distribution

In this section, we analyze trace-norm regularized learning when the sampling distribution is not uniform. That is, when there is some, known or unknown, non-uniform distribution $\mathcal{D}$ over entries of the matrix Y (i.e. over index pairs (i,j)) and our sample S is sampled i.i.d. from $\mathcal{D}$. Our objective is to get low average error with respect to the distribution $\mathcal{D}$. That is, we measure generalization performance in terms of the weighted sum-squared-error:

$$\|X - Y\|_{\mathcal{D}}^2 \quad = \quad \mathbf{E}_{(i,j)\sim\mathcal{D}}\left[(X_{ij} - Y_{ij})^2\right] = \sum_{ij} \mathcal{D}(i,j)(X_{ij} - Y_{ij})^2. \tag{6}$$

We first point out that when using the rank for complexity control, i.e. when minimizing the training error subject to a low-rank constraint, non-uniformity does *not* pose a problem. The same generalization and learning guarantees that can be obtained in the uniform case, also hold under an arbitrary distribution $\mathcal{D}$. In particular, if there is some low-rank X^* such that $\|X^* - Y\|_{\mathcal{D}}^2$ is small, then $\tilde{O}(rank(X^*)(n+m))$ samples are enough in order to learn (by minimizing training error subject to a rank constraint) a matrix X with $\|X - Y\|_{\mathcal{D}}^2$ almost as small as $\|X^* - Y\|_{\mathcal{D}}^2$ [16].

However, the same does not hold when learning using the trace-norm. To see this, consider an orthogonal rank-k square $n \times n$ matrix, and a sampling distribution which is uniform over an $n_A \times n_A$ sub-matrix A, with $n_A = n^a$. That is, the row (e.g. "user") is selected uniformly among the first n_A rows, and the column (e.g. "movie") is selected uniformly among the first n_A columns. We will use A to denote the subset of entries in the submatrix, i.e. $A = \{(i,j) | 1 \leq i,j \leq n_A\}$. For any sample S, we have:

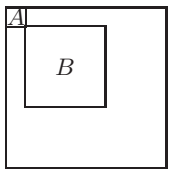
$$tc(Y_S) = \frac{\|Y_S\|_{\text{tr}}^2}{n^2} \leq \frac{\|Y_S\|_{\text{F}}^2 \, rank(Y_S)}{n^2} \leq \frac{|S| n^a}{n^2} = \frac{|S|}{n^{2-a}}, \tag{7}$$

where we again take the entries in Y to be of unit magnitude. In the second inequality above we use the fact that Y_S is zero outside of A, and so we can bound the rank of Y_S by the dimensionality $n_A = n^a$ of A.

Setting $a < 1$, we see that we can shatter any sample of size[1] $kn^{2-a} = \tilde{\omega}(n)$ with a matrix X for which $tc(X) < k$. When $a \leq 1/2$, the total number of entries in A is less than n. In this case $\tilde{O}(n)$ observations are enough in order to memorize[2] Y_A. But when $1/2 < a < 1$, with $\tilde{O}(n)$ observations, restricting to even $tc(X) < 1$, we can neither learn Y, since we can shatter Y_S, nor memorize it. For example, when $a = 2/3$ and so $n_A = n^{2/3}$, we need roughly $n^{4/3}$ to start learning by constraining $tc(X)$ to a constant — the same as we would need in order to memorize Y_A. This is a factor of $n^{1/3}$ greater than the sample size needed to learn a matrix with constant $tc(X)$ in the uniform case.

The above arguments establish that restricting the complexity to $tc(X) < k$ might not lead to generalization with $\tilde{O}(kn)$ samples in the non-uniform case. But does this mean that we cannot learn a rank-k matrix by minimizing the trace-norm using $\tilde{O}(kn)$ samples when the sampling distribution is concentrated on a small submatrix? Of course this is not the case. Since the samples are uniform on a small submatrix, we can just think of the submatrix A as our entire space. The target matrix still has low rank, even when restricted to A, and we are back in the uniform sampling scenario. The only issue here is that $tc(X) \leq k$, i.e. $\|X\|_{\text{tr}} \leq n\sqrt{k}$, is the right constraint in the uniform observation scenario. When samples are concentrated in n_A, we actually need to restrict to a much smaller trace norm, $\|X\|_{\text{tr}} \leq n^a\sqrt{k}$, which will allow learning with $\tilde{O}(kn^a)$ samples.

We can, however, modify the example and construct a sampling distribution under which $\Omega(n^{4/3})$ samples are required in order to learn even an "orthogonal" low-rank matrix, no matter what constraint is placed on the trace-norm. This is a significantly large sample complexity than $\tilde{O}(kn)$, which is what we would expect, and what is required for learning by constraining the rank directly.



To do so, consider another submatrix B of size $n_B \times n_B$ with $n_B = n/2$, such that the rows and columns of A and B do not overlap (see figure). Now, consider a sampling distribution $\mathcal{D}$ which is uniform over A with probability half, and uniform over B with probability half. Consider fitting a noisy matrix $Y = X^* + \text{noise}$ where X^* is "orthogonal" rank-k. In order to fit on B, we need to allow a trace-norm of at least $\|X^*_B\|_{\text{tr}} = \frac{n}{2}\sqrt{k}$, i.e. allow $tc(X) = k/4$. But as discussed above, with such a generous constraint on the trace-norm, we will be able to shatter $S \subset A$ whenever $|S \cap A| = |S|/2 \leq kn^{2-a}/4$. Since there is no overlap in rows and columns, and so values in the sub-matrices A and B are independent, shattering $S \cap A$ means we cannot hope to learn in A. Setting a=2/3 as before, with $o(n^{4/3})$ samples, we cannot learn in A and B jointly: either we constrain to a trace-norm which is too low to fit X^*_B (we under-fit on B), or we allow a trace-norm which is high enough to overfit $Y_{S\cap A}$. In any case, we will make errors on at least half the mass of $\mathcal{D}$.[3]

Empirical Example

Let us consider a simple simulation experiment that will help us illustrates this phenomenon. Consider a simple synthetic example, where we used $n_A = 300$ and $n_B = 4700$, with an orthogonal rank-2 matrix X^* and $Y = X^* + \mathcal{N}(0, 1)$ (in case of repeated entries, the noise is independent for each appearance in the sample). The training sample size was also set to $|S|$=140,000.

The three curves of Fig. 1 measure the excess (test) error $\|X - X^*\|^2_{\mathcal{D}} = \|X - Y\|^2_{\mathcal{D}} - \|Y - X^*\|^2_{\mathcal{D}}$ of the learned model, as well as the error contribution from A and from B, as a function of the constraint on $tc(X)$, for the sampling distribution discussed above and a specific sample size. As can be seen, although it is possible to constrain $tc(X)$ so as to achieve squared-error of less than 0.8 on B, this constraint is too lax for A and allows for over-fitting. Constraining $tc(X)$ so as to avoid overfitting A (achieving almost zero excess test error), leads to a suboptimal fit on B.

[1] Recall that $f(n) = \tilde{\omega}(g(n))$ iff for all p, $\frac{g(n)\log^p g(n)}{f(n)} \to 0$.

[2] The algorithm saw all (or most) entries of the matrix and does not need to predict any unobserved entries.

[3] More accurately, if we do allow high enough trace-norm to fit B, and $|S| = o(n^{4/3})$, then the "cost" of overfitting $Y_{S\cap A}$ is negligible compared to the cost of fitting X^*_B. For large enough n, we would be tempted to very slightly deteriorate the fit of X^*_B in order to "free up" enough trace-norm and completely overfit $Y_{S\cap A}$.

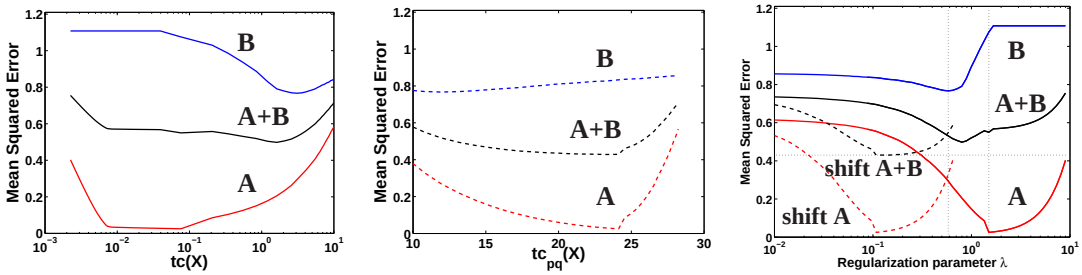


Figure 1: **Left**: Mean squared error (MSE) of the learned model as a function of the constraint on $tc(X)$ (left), $tc_{pq}(X)$ (middle). **Right**: The solid curves show the optimum of the mean squared error objective (9) (unweighted trace-norm), as a function of the regularization parameter λ. The dashed curves display a weighted trace-norm. The black (middle) curve is the overall MSE error, the red (bottom) curve measures only the contribution from A, and the blue (top) curve measures only the contribution from B.

Penalty Formulation

Until now we discussed learning by constraining the trace-norm, i.e. using the formulation (2). It is also insightful to consider the penalty view (1), i.e. learning by minimizing

$$\min_X \|Y_S - X_S\|_{\mathrm{F}}^2 + \lambda \|X\|_{\mathrm{tr}} . \tag{8}$$

First observe that the characterization (3) allows us to decompose $\|X\|_{\mathrm{tr}} = \|X_A\|_{\mathrm{tr}} + \|X_B\|_{\mathrm{tr}}$, where w.l.o.g. we take all columns of U and V outside A and B to be zero. Since we also have $\|Y_S - X_S\|_{\mathrm{F}}^2 = \|Y_{A\cap S} - X_{A\cap S}\|_{\mathrm{F}}^2 + \|Y_{B\cap S} - X_{B\cap S}\|_{\mathrm{F}}^2$, we can decompose the training objective (8) as:

$$\begin{aligned}\|Y_S - X_S\|_{\mathrm{F}}^2 + \lambda \|X\|_{\mathrm{tr}} &= (\|Y_{A\cap S} - X_{A\cap S}\|_{\mathrm{F}}^2 + \lambda \|X_A\|_{\mathrm{tr}}) + (\|Y_{B\cap S} - X_{B\cap S}\|_{\mathrm{F}}^2 + \lambda \|X_B\|_{\mathrm{tr}}) \\ &= \left(\|Y_{A\cap S} - X_{A\cap S}\|_{\mathrm{F}}^2 + \lambda n_A \sqrt{tc_A(X_A)}\right) + \left(\|Y_{B\cap S} - X_{B\cap S}\|_{\mathrm{F}}^2 + \lambda n_B \sqrt{tc_B(X_B)}\right), \end{aligned} \tag{9}$$

where $tc_A(X_A) = \|X_A\|_{\mathrm{tr}}^2 / n_A^2$ (and similarly $tc_B(X_B)$) refers to the complexity measure $tc(\cdot)$ measured relative to the size of A (similarly B). We see that the training objective decomposes to objectives over A and B. Each one of these corresponds to a trace-norm regularized learning problem, under a uniform sampling distribution (in the corresponding submatrix) of a noisy low-rank "orthogonal" matrix, and can therefor be learned with $\tilde{O}(kn_A)$ and $\tilde{O}(kn_B)$ samples respectively. In other words, $\tilde{O}(kn)$ samples should be enough to learn both inside A and inside B.

However, the regularization tradeoff parameter λ compounds the two problems. When the objective is expressed in terms of $tc(\cdot)$, as in (9), the regularization tradeoff is scaled differently in each part of the training objective. With $\tilde{O}(kn)$ samples, it is possible to learn in A with some setting of λ, and it is possible to learn in B with some other setting of λ, but from the discussion above we learn that no single value of λ will allow learning in both A and B. Either λ is too high yielding too strict regularization in B, so learning on B is not possible, perhaps since it is scaled by $n_B \gg n_A$. Or λ is too small and does not provide enough regularization in A.

Returning to our simulation experiment, the solid curves of Fig. 1, right panel, show the excess test error for the minimizer of the training objective (9), as a function of the regularization tradeoff parameter λ. Note that these are essentially the same curves as displayed in Fig. 1, except the path of regularized solutions is now parameterized by λ rather than by the bound on $tc(X)$. Not surprisingly, we see the same phenomena: different values of λ are required for optimal learning on A and on B. Forcing the same λ on both parts of the training objective (9) yields a deterioration in the generalization performance.

4 Weighted Trace Norm

The decomposition (9) and the discussion in the previous section suggests weighting the trace-norm by the frequency of rows and columns. For a sampling distribution $\mathcal{D}$, denote by $p(i)$ the row marginal, i.e. the probability of observing row i, and similarly denote by $q(j)$ the column marginal. We propose using the following weighted version of the trace-norm as a regularizer:

$$\|X\|_{\mathrm{tr}(p,q)} = \|\mathrm{diag}(\sqrt{p}) X \mathrm{diag}(\sqrt{q})\|_{\mathrm{tr}} = \min_{X=U'V} \frac{1}{2}\Big(\sum_i p(i) \|U_i\|^2 + \sum_j q(j), \|V_j\|^2\Big) \tag{10}$$

where diag($\sqrt{p}$) is a diagonal matrix with $\sqrt{p(i)}$ on its diagonal (similarly diag($\sqrt{q}$)). The corresponding normalized complexity measure is given by $tc_{pq}(X) = \|X\|^2_{\mathrm{tr}(p,q)}$. Note that for a uniform distribution we have that $tc_{pq}(X) = tc(X)$. Furthermore, it is easy to verify that for an "orthogonal" rank-k matrix X we have $tc_{pq}(X) = k$ for *any* sampling distribution.

Equipped with the weighted trace-norm as a regularizer, let us revisit the problematic sampling distribution studied in the previous Section. In order to fit the "orthogonal" rank-k X^*, we need a weighted trace-norm of $\|X^*\|_{\mathrm{tr}(p,q)} = \sqrt{tc_{pq}(X)} = \sqrt{k}$. How large a sample $S \cap A$ can we now shatter using such a weighted trace-norm? We can shatter a sample if $\|Y_{S\cap A}\|_{\mathrm{tr}} \leq \sqrt{k}$. We can calculate:

$$\|Y_{S\cap A}\|_{\mathrm{tr}(p,q)} = \|Y_{S\cap A}\|_{\mathrm{tr}} / (2n_A) \leq \sqrt{|S \cap A| n_A}/(2n_A) = \sqrt{|S|/(8n_A)}. \tag{11}$$

That is, we can shatter a sample of size up to $|S| = 8kn_A < 8kn$. The calculation for B is identical. It seems that now, with a fixed constraint on the weighted trace-norm, we have enough capacity to both fit X^*, and with $\tilde{O}(kn)$ samples, avoid overfitting on A.

Returning to the penalization view (2) we can again decompose the training objective as:

$$\begin{aligned} &\|Y_S - X_S\|^2_{\mathrm{F}} + \lambda \|X\|_{\mathrm{tr}(p,q)} = \\ &= \left(\|Y_{A\cap S} - X_{A\cap S}\|^2_{\mathrm{F}} + \lambda/2\sqrt{tc_A(X_A)}\right) + \left(\|Y_{B\cap S} - X_{B\cap S}\|^2_{\mathrm{F}} + \lambda/2\sqrt{tc_B(X_B)}\right) \end{aligned} \tag{12}$$

avoiding the scaling by the block sizes which we encountered in (9).

Returning to the synthetic experiments of Fig. 1 (right panel), and comparing (9) with (12), we see that introducing the weighting corresponds to a relative change of n_A/n_B in the correspondence of the regularization tradeoff parameters used for A and for B. This corresponds to a shift of $\log \frac{n_A}{n_B}$ in the log-domain used in the figure. Shifting the solid red (bottom) curve by this amount yields the dashed red (bottom) curve. The solid blue (top) curve and the dashed red (bottom) curve thus represent the excess error on B and on A when the weighted trace norm is used, i.e. the training objective (12) is minimized. The dashed black (middle) curve is the overall excess error when using this training objective. As can be seen, the weighting aligns the excess errors on A and on B much better, and yields a lower overall error. The weighted trace-norm achieves the lowest MSE of 0.4301 with corresponding $\lambda = 0.11$. This is compared to the lowest MSE of 0.4981 with $\lambda = 0.80$, achieved by the unweighted trace-norm.

It is also interesting to observe that the weighted trace-norm outperforms its unweighted counterpart for a wide range of regularization parameters $\lambda \in [0.01; 0.6]$. This may also suggest that in practice, particularly when working with large and imbalanced datasets, it may be easier to search for regularization parameters using weighted trace-norm.

Finally, Fig. 1, right panel, also suggests that the optimal shift might actually be smaller than n_A/n_B. We can consider a smaller shift by using the partially-weighted trace-norm:

$$\|X\|_{\mathrm{tr}(p,q,\alpha)} = \left\|\mathrm{diag}(p^{\alpha/2}) X \mathrm{diag}(q^{\alpha/2})\right\|_{\mathrm{tr}} = \min_{X=U^\top V} \frac{1}{2}\left(\sum_i p(i)^\alpha \|U_i\|^2 + \sum_j q(j)^\alpha \|V_j\|^2\right).$$

and the corresponding normalized complexity measure $tc^\alpha(X) = \|X\|^2_{\mathrm{tr}(p^\alpha/n^{1-\alpha}, q^\alpha/m^{1-\alpha})}$.

Other Weightings and Bayesian Perspective

The weighted trace-norm motivated by the analysis here (with $\alpha = 1$) implies that the frequent users (equivalently movies) get regularized much stronger than the rare users (equivalently movies). This might at first seem quite counter-intuitive as the natural weighting might seem to be the opposite. Indeed, Weimer *et al.* [21] speculated that with a uniform weighting ($\alpha = 0$) frequent users are regularized too heavily compared to infrequent users, and so suggested regularizing frequent users (and movies) with a lower weight, corresponding to $\alpha = -1$. Although this might seem natural, we saw here that the reverse is actually true – the Weimer *et al.* weighting ($\alpha = -1$) would only make things worse. Indeed, given the analysis here, Weimer *et al.* actually observed a deterioration in prediction quality when using their weighting. This is also demonstrated in the experiments on the Netflix data in Section 6.

The weighted regularization motivated here (with $\alpha = 1$) is also quite unusual from Bayesian perspective. The trace-norm can be viewed as a negative-log-prior for the Probabilistic Matrix Factorization model [15], where entries of U, V are taken to be i.i.d. Gaussian. The two terms of (8) can then be interpreted as a log-likelihood and log-prior, and minimizing (8) corresponds to finding the MAP parameters. Introducing weighting (with $\alpha = 1$) effectively states that the effect of the prior becomes *stronger* as we observe more data. Yet, our analysis strongly suggest that in non-uniform setting, such "unorthodox" regularization is crucial for achieving good generalization performance.

5 Practical Implementation

When dealing with large datasets, such as the Netflix data, the most practical way to fit trace-norm regularized models is through stochastic gradient descent [15, 8]. Let $n_i = \sum_j S_{ij}$ and $m_j = \sum_i S_{ij}$ denote the number of observed ratings for user i and movie j respectively. The training objective using a partially-weighted trace-norm 10 can be written as:

$$\sum_{\{i,j\}\in S} \left(\left(Y_{ij} - U_i^\top V_j\right)^2 + \frac{\lambda}{2} \left(\frac{p(i)^\alpha}{n_i} \left\|U_i\right\|^2 + \frac{q(j)^\alpha}{m_j} \left\|V_j\right\|^2 \right) \right),$$

where $U \in \mathbb{R}^{k\times n}$ and $V \in \mathbb{R}^{k\times m}$. We can optimize this objective using stochastic gradient descent by picking one training pair (i, j) at random at each iteration, and taking a step in the direction opposite the gradient of the term corresponding to the chosen (i, j).

Note that even though the objective (13) as a function of U and V is non-convex, there are no non-global local minima if we set k to be large enough, i.e. $k > \min(n, m)$ [2]. However, in practice using very large values of k becomes computationally expensive. Instead, we consider truncated trace-norm minimization by restricting k to smaller values. In the next section we demonstrate that even when using truncated trace-norm, its weighted version significantly improves model's prediction performance.

In our experiments, we also replace unknown row $p(i)$ and column $q(j)$ marginals in (13) by their empirical estimates $\hat{p}(i) = {}^{n_i}/_{|S|}$ and $\hat{q}(j) = {}^{m_j}/_{|S|}$. This results in the following objective:

$$\sum_{\{i,j\}\in S} \left(\left(Y_{ij} - U_i^\top V_j\right)^2 + \frac{\lambda}{2|S|} \left(n_i^{\alpha-1} \left\|U_i\right\|^2 + m_j^{\alpha-1} \left\|V_j\right\|^2 \right) \right). \tag{13}$$

Setting $\alpha = 1$, corresponding to the weighted trace-norm (10), results in stochastic gradient updates that do not involve the row and column counts at all and are in some sense the simplest. Strangely, and likely originating as a "bug" in calculating the stochastic gradients by one of the participants, these steps match the stochastic training used by many practitioners on the Netflix dataset, without explicitly considering the weighted trace-norm [8, 19, 15].

6 Experimental results

We tested the weighted trace-norm on the Netflix dataset, which is the largest publicly available collaborative filtering dataset. The training set contains 100,480,507 ratings from 480,189 anonymous users on 17,770 movie titles. Netflix also provides qualification set, containing 1,408,395 ratings, out of which we set aside 100,000 ratings for validation. The "qualification set" pairs were selected by Netflix from the most recent ratings for a subset of the users. Due to the special selection scheme, ratings from users with few ratings are overrepresented in the qualification set, relative to the training set. To be able to report results where the train and test sampling distributions are the same, we also created a "test set" by randomly selecting and removing 100,000 ratings from the training set. All ratings were normalized to be zero-mean by subtracting 3.6. The dataset is very imbalanced: it includes users with over 10,000 ratings as well as users who rated fewer than 5 movies.

For various values of α, we learned a factorization $U^\top V$ with $k = 30$ and with $k = 100$ dimensions (factors) using stochastic gradient descent as in (13). For each value of α and k we selected the regularization tradeoff λ by minimizing the error on the 100,000 qualification set examples set aside for validation. Results on both the Netflix qualification set and on the test set we created are reported in Table 1. Recall that the sampling distribution of the "test set" matches that of the training data, while the qualification set is sampled differently, explaining the large difference in generalization between the two.

Table 1: Root Mean Squared Error (RMSE) on the Netflix qualification set and on a test set that was held out from the training data, for training by minimizing (13). We report $\lambda/|S|$ minimizing the error on the validation set (held out from the qualification set), qualification and test errors using this tradeoff, and $tc^{\alpha}(X)$ at the optimum. Last row: training by regularizing the max-norm.

α	k	$\lambda/\lvert S\rvert$	$tc^{\alpha}(X)$	Test	Qual	k	$\lambda/\lvert S\rvert$	$tc^{\alpha}(X)$	Test	Qual
1	30	0.05	4.34	0.7607	0.9105	100	0.08	5.47	0.7412	0.9071
0.9	30	0.07	4.27	0.7573	0.9091	100	0.1	5.23	0.7389	0.9062
0.75	30	0.2	5.04	0.7723	0.9128	100	0.3	6.24	0.7491	0.9098
0.5	30	0.5	7.32	0.7823	0.9159	100	0.8	9.65	0.7613	0.9127
0	30	2.5	10.36	0.7889	0.9235	100	3.0	21.23	0.7667	0.9203
-1	30	450	11.41	0.7913	0.9256	100	700	23.31	0.7713	0.9221
$\lVert X\rVert_{\max}$	30	$mc(X) = 5.06$		0.7692	0.9131	100	$mc(X) = 5.77$		0.7432	0.9092

For both $k = 30$ and $k = 100$, the weighted trace-norm ($\alpha = 1$) significantly outperformed the unweighted trace-norm ($\alpha = 0$). Interestingly, the optimal weighting (setting of α) was a bit lower then, but very close to $\alpha = 1$. For completeness, we also evaluated the weighting suggested by Weimer *et al.* [21], corresponding to $\alpha = -1$. Unsurprising, given our analysis, this seemingly intuitive weighting hurts predictive performance.

For both $k = 30$ and $k = 100$, we also observed that for the weighted trace-norm ($\alpha = 1$) good generalization is possible with a wide range of λ settings, while for the unweighted trace-norm ($\alpha = 0$), the results were much more sensitive to the setting of λ. This confirms our previous results on the synthetic experiment and strongly suggests that it may be far easier to search for regularization parameters using the weighted trace-norm.

Comparison with the Max-Norm

We also compared the predictive performance on Netflix to predictions based on max-norm regularization. The max-norm is defined as:

$$\lVert X\rVert_{\max} = \min_{X=U'V} \frac{1}{2}(\max_i \lVert U_i\rVert^2 + \max_j \lVert V_j\rVert^2). \tag{14}$$

Similarly to the rank, but unlike the trace-norm, generalization and learning guarantees based on the max-norm hold also under an arbitrary, non-uniform, sampling distribution. Specifically, defining $mc(X) = \lVert X\rVert^2_{\max}$ (no normalization is necessary here), $\tilde{O}(mc(X)(n+m))$ samples are enough for generalization w.r.t. any sampling distribution (just like the rank) [18]. This suggests that perhaps the max-norm can be used as an alternative factorization-regularization in the presence of non-uniform sampling. Indeed, as evident in Table 1, max-norm based regularization does perform much better then the unweighted trace-norm. The differences between the max-norm and the weighted trace-norm are small, but it seems that using the weighted trace-norm is slightly but consistently better.

7 Summary

In this paper we showed both analytically and empirically that under non-uniform sampling, trace-norm regularization can lead to significant performance deterioration and an increase in sample complexity. Our analytic analysis suggests a non-intuitive weighting for the trace-norm in order to correct the problem. Our results on both synthetic and on the highly imbalanced Netflix datasets further demonstrate that the weighted trace-norm yields significant improvements in prediction quality.

In terms of optimization, we focused on stochastic gradient descent,both since it is a simple and practical method for very large-scale trace-norm optimization [15, 8], and since the weighting was originally stumbled upon through this optimization approach. However, most recently proposed methods for trace-norm optimization (e.g. [3, 10, 9, 11, 20]) can also be easily modified for the weighted trace-norm.

We hope that the weighted trace-norm, and the discussions in Sections 3 and 4, will be helpful in deriving theoretical learning guarantees for arbitrary non-uniform sampling distributions, both in the form of generalization error bounds as in [18], and generalizing the compressed-sensing inspired work on recovery of noisy low-rank matrices as in [4, 13].

Acknowledgments RS is supported by NSERC, Shell, and NTT Communication Sciences Laboratory.

References

[1] J. Abernethy, F. Bach, T. Evgeniou, and J.P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research*, 10:803–826, 2009.

[2] S. Burer and R.D.C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

[3] J.F. Cai, E.J. Candès, and Z. Shen. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM Journal on Optimization*, 20:1956, 2010.

[4] E.J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE (to appear)*, 2009.

[5] E.J. Candes and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9, 2009.

[6] E.J. Candes and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inform. Theory (to appear)*, 2009.

[7] M. Fazel, H. Hindi, and S.P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings American Control Conference*, volume 6, 2001.

[8] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *ACM SIGKDD*, pages 426–434, 2008.

[9] Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2009.

[10] S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, pages 1–33, 2009.

[11] R. Mazumder, T. Hastie, and R. Tibshirani. Spectral Regularization Algorithms for Learning Large Incomplete Matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

[12] R. Meka, P. Jain, and I. S. Dhillon. Matrix completion from power-law distributed samples. In *Advances in Neural Information Processing Systems*, volume 21, 2009.

[13] B. Recht. A simpler approach to matrix completion. preprint, available from author's webpage, 2009.

[14] J.D.M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, page 719, 2005.

[15] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[16] N. Srebro, N. Alon, and T. Jaakkola. Generalization error bounds for collaborative prediction with low-rank matrices. In *Advances In Neural Information Processing Systems 17*, 2005.

[17] N. Srebro, J. Rennie, and T. Jaakkola. Maximum margin matrix factorization. In *Advances In Neural Information Processing Systems 17*, 2005.

[18] N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In *COLT*, 2005.

[19] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656, 2009.

[20] R. Tomioka, T. Suzuki, M. Sugiyama, and H. Kashima. A fast augmented lagrangian algorithm for learning low-rank matrices. In *ICML*, pages 1087–1094, 2010.

[21] M. Weimer, A. Karatzoglou, and A. Smola. Improving maximum margin matrix factorization. *Machine Learning*, 72(3):263–276, 2008.