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

Ruslan Salakhutdinov

Brain and Cognitive Sciences and CSAIL, Massachusetts Institute of Technology

RSALAKHU@MIT.EDU

Nathan Srebro

Toyota Technological Institute–Chicago

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. We introduce a weighted version of the trace-norm regularizer that works well also with non-uniform sampling. Our experimental results demonstrate 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 (Fazel et al., 2001; Candes & Tao, 2009) and in terms of a regularized infinite factor model with connections to large-margin norm-regularized learning (Srebro et al., 2005b; Bach, 2008; Abernethy et al., 2009; Salakhutdinov & Mnih, 2008).

Current theoretical guarantees on using the trace-norm for matrix completion all assume a uniform sampling distribution over entries of the matrix (Srebro & Shraibman, 2005; Candes & Tao, 2009; Candes & Recht, 2009; Candes & Tao, 2009; Recht, 2009). 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 in any actual collaborative filtering application, some users are much more active than others and some movies are rated by many people while others are much less likely to be rated.

In Section 3 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.

In Section 4 we suggest a correction to the trace-norm regularizer, which we call the *weighted* trace-norm, that takes into account the sampling distribution. This correction is motivated by our analytic analysis and we discuss how it corrects the problems that the unweighted trace-norm has with non-uniform sampling. We then show how the weighted trace-norm indeed yields a significant improvement on the (highly non-uniformly sampled) Netflix dataset.

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:

$$\min_X \|X_S - Y_S\|_F^2 + \lambda c(X) \quad (1)$$

or:

$$\min_{c(X) \leq C} \|X_S - Y_S\|_F^2, \quad (2)$$

where Y_S , and similarly X_S , denotes the matrix “masked” by S :

$$(Y_S)_{i,j} = \begin{cases} Y_{i,j} & \text{if } (i,j) \in S \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

For now we ignore possible repeated entries in S . We will also assume that $n \leq m$ 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 at our convenience.

2.1. Low Rank Factorization

A basic measure of complexity is the rank of X , corresponding to the minimal dimensionality k such that $X = U^T V$ 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. Training such a model amounts to finding the best rank- k approximation to the observed target matrix Y under the given loss function. 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.

2.2. Trace-norm Regularization

Lately, methods regularizing the *norm* of the factorization $U^T V$, rather than its dimensionality, have been advocated and were shown to enjoy considerable empirical success (Rennie & Srebro, 2005; Salakhutdinov & Mnih, 2008). This is captured by 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 (Fazel et al., 2001):

$$\|X\|_{\text{tr}} = \min_{X=U^T V} \frac{1}{2} (\|U\|_F^2 + \|V\|_F^2). \quad (4)$$

Note that the dimensionality of U and V in (4) 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.

2.3. Scaling of the Trace-norm

It will be useful for us to consider the scaling of the trace-norm with the size of the matrix X . This will

allow us, for example, to understand the magnitude of the bound C we can expect to put on the trace-norm in the formulation (2).

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 inherently 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:

$$\|X\|_F \leq \|X\|_{\text{tr}} \leq \|X\|_F \sqrt{\text{rank}(X)} \leq n \|X\|_F, \quad (5)$$

where in the second inequality we used the fact that the number of non-zero singular values is equal to the rank. 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 magnitude¹, e.g. ± 1 , 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:

$$tc(X) = \frac{\|X\|_{\text{tr}}^2}{nm}, \quad (6)$$

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), in which case $\|X\|_F = \sqrt{nm}$, we have:

$$1 \leq tc(X) \leq \text{rank}(X) \leq n. \quad (7)$$

To further understand the trace-norm complexity control, consider “orthogonal” low-rank matrices $U \in \mathbb{R}^{k \times n}$ and $V \in \mathbb{R}^{k \times m}$, such that $Y = U^T V$ and where the entries of U and V are i.i.d. $\mathcal{N}(0, 1/\sqrt{k})^2$. The matrix Y is then of rank k , with each entry having zero mean and unit variance (magnitude). Its Frobenius norm is tightly concentrated at $\|Y\|_F = \sqrt{nm}$.

¹Any other constant magnitude will only result in some constant scaling

²The important issue here is the orthogonality and the norm uniformity, not the randomness. But we find it easier to think of the orthogonality in terms of an i.i.d. random model.

Since rows of U and V are orthogonal, this is essentially the singular value decomposition, with all k singular values being equal to $\sqrt{nm/k}$. We thus have $tc(X) = k$. And so at least in the orthogonal case, $tc(X) = rank(X)$.

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. Such learning guarantees were mostly discussed in the context of Lipschitz continuous loss functions (i.e. functions with a bounded first derivative), rather than the squared loss. The squared loss has a bounded second derivative rather than bounded first derivative and so requires somewhat different technical tools. Nevertheless, the main thrust of the results is still valid.

For Lipschitz continuous loss functions, 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^* (Srebro et al., 2005a). 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^* (Srebro & Shraibman, 2005). In these bounds $tc(X)$ plays precisely the same role as the rank, up to logarithmic factors.

Without getting into the technical tools required to rigorously establish the above sample complexity guarantees, it is useful to understand them at a more abstract level. In order to understand the guarantees for low-rank learning, it is enough to consider the number of parameters in the rank- k factorization $X = U^T V$. 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). And so we 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 $\|Y_S\|_F = \sqrt{|S|} = \sqrt{Cn}$ (Recall that Y_S is defined to be zero outside S). From (5) we therefore have: $\|Y_S\|_{tr} \leq \sqrt{Cn} \cdot \sqrt{n} = \sqrt{Cn}$ 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’t 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 $tc(X) \leq C = tc(X^*)$, and that if we have more than this many samples we can start learning. For our purposes here, we will mostly just make use of non-learnability arguments of this form: if we can shatter a random sample of size $|S|$ with a matrix X have the same complexity (e.g. trace-norm) as our target matrix X^* , we cannot hope to learn without a larger sample.

3. Trace-Norm Under a Non-Uniform Distribution

In this section, we will 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} . Of course, if \mathcal{D} concentrates on only a small subset of the matrix, we have no hope of recovering rows and columns of Y on which we have zero probability of seeing an observation. Instead, our objective here, as is typically the case in learning under an arbitrary distribution, is to get low average error with respect to the same distribution \mathcal{D} . That is, we measure generalization performance in terms of the weighted sum-squared-error:

$$\begin{aligned} \|X - Y\|_{\mathcal{D}}^2 &= \mathbf{E}_{(i,j) \sim \mathcal{D}} [(X_{ij} - Y_{ij})^2] \\ &= \sum_{ij} \mathcal{D}(i, j) (X_{ij} - Y_{ij})^2. \end{aligned} \quad (8)$$

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 $\|X^* - Y\|_{\mathcal{D}}^2$ (Srebro et al., 2005a)³.

However, the same does not hold when learning us-

³Actually, this is shown only for Lipschitz continuous loss functions, and not for the squared-loss, but at the very least this holds if X is appropriately clipped. Since for-

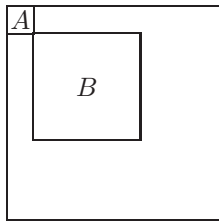


Figure 1. The two submatrices A of size $n_A = n^a$ and B of size $n_B = n/2$.

ing 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 $n_A \times n_A$ submatrix A , with $n_A = n^a$ (see Fig. 1). 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. $A = \{(i, j) | 1 \leq i, j \leq n_A\}$, rather than the matrix itself, and so we can say that \mathcal{D} is uniform on A . For any sample S , we have:

$$\begin{aligned} tc(Y_S) &= \frac{\|Y_S\|_{\text{tr}}^2}{n^2} \leq \frac{\|Y_S\|_{\text{F}}^2 \text{rank}(Y_S)}{n^2} \\ &\leq \frac{|S|n^a}{n^2} = \frac{|S|}{n^{2-a}}, \end{aligned} \quad (9)$$

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 size⁴ $kn^{2-a} = \tilde{\omega}(n)$ 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 , and so $\tilde{O}(n)$ observations are enough in order to memorize 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? General guarantees are not the focus of this paper, we rather view this statement only as an indicative statement without stating it rigorously.

⁴Recall that $f(n) = \tilde{\omega}(g(n))$ is the same as $g(n) = \tilde{o}(f(n))$ and means that for all p we have $\frac{g(n) \log^p g(n)}{f(n)} \rightarrow 0$.

trix 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.

It is, however, easy to modify the above 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 of B do not overlap (Fig. 1). 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 $|S \cap A| = |S|/2 \leq k/4n^{2-a}$. 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, it seems that with $o(n^{4/3})$ samples, we cannot learn in both A and B : 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 $Y_{S \cap A}$. Either way, we will make errors on at least half the mass of \mathcal{D} .⁵

Figure 2, left panel, precisely illustrates this phenomenon on a simulation experiment. For this synthetic example, we used $n_A = 300$ and $n_B = 4700$,

⁵To make the above argument more precise, we should note that 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 $Y_{S \cap A}$.

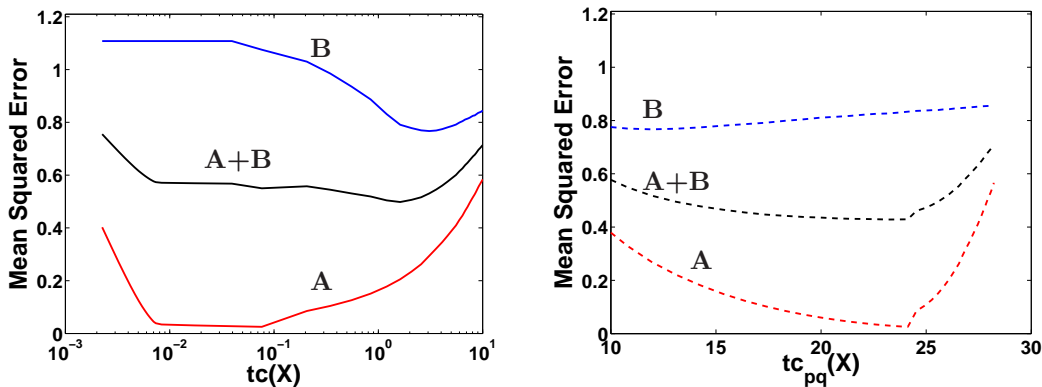


Figure 2. Mean squared error (MSE) of the learned model as a function of the constraint on $tc(X)$ (left) and $tc_{pq}(X)$ (right). The black (middle) curve is the overall MSE error, the red (bottom) curve measures only the contribution from A , and the blue (top) curve measures only the contribution from B .

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. 2 measure the excess (test) error $\|X - X^*\|_{\mathcal{D}}^2 = \|X - Y\|_{\mathcal{D}}^2 - \|Y - X^*\|_{\mathcal{D}}^2$ 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 .

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

$$\min_X \|Y_S - X_S\|_F^2 + \lambda \|X\|_{\text{tr}}. \quad (10)$$

First observe that the characterization (4) allows us to decompose $\|X\|_{\text{tr}} = \|X_A\|_{\text{tr}} + \|X_B\|_{\text{tr}}$, where w.l.o.g. we take all columns of U and V outside A and B to be zero. Since we also have $\|Y_S - X_S\|_F^2 = \|Y_{A \cap S} - X_{A \cap S}\|_F^2 + \|Y_{B \cap S} - X_{B \cap S}\|_F^2$, we can decompose the training objective (10) as:

$$\begin{aligned} & \|Y_S - X_S\|_F^2 + \lambda \|X\|_{\text{tr}} \\ &= (\|Y_{A \cap S} - X_{A \cap S}\|_F^2 + \lambda \|X_A\|_{\text{tr}}) \\ & \quad + (\|Y_{B \cap S} - X_{B \cap S}\|_F^2 + \lambda \|X_B\|_{\text{tr}}) \\ &= \left(\|Y_{A \cap S} - X_{A \cap S}\|_F^2 + \lambda n_A \sqrt{tc_A(X_A)} \right) \\ & \quad + \left(\|Y_{B \cap S} - X_{B \cap S}\|_F^2 + \lambda n_B \sqrt{tc_B(X_B)} \right), \quad (11) \end{aligned}$$

where $tc_A(X_A) = \|X_A\|_{\text{tr}}^2 / n_A^2$ (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 a trace-norm regularized problem in A and a trace-norm regularized problem in B . Each one of these problems is a trace-norm regularized learning problem, under a uniform sampling distribution (in the corresponding submatrix) of a noisy low-rank “orthogonal” matrix, and can therefore 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 (11), 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. 3 show the excess test error for the minimizer of the training objective (11), as a function of the regularization tradeoff parameter λ . Note that these are essentially the same curves as displayed in Fig. 2, 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 (11) yields a deterioration in the generalization performance.

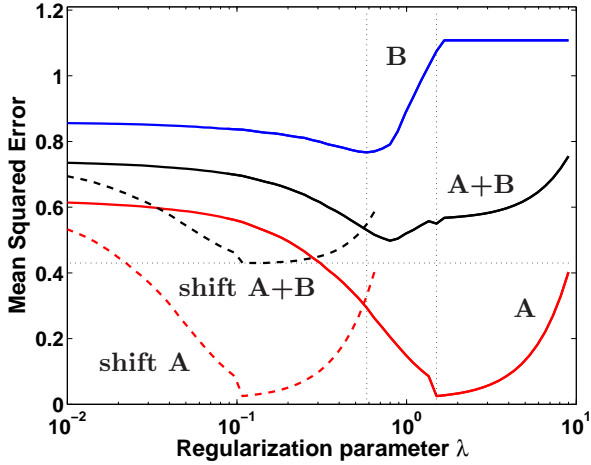


Figure 3. The solid curves show the optimum of the mean squared error objective (11) (unweighted trace-norm), as a function of the regularization parameter λ . The dashed curves display a weighted trace-norm.

4. Weighted Trace Norm

The decomposition (11) 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 weighted version of the trace-norm as a regularizer:

$$\begin{aligned} \|X\|_{\text{tr}(p,q)} &= \|\text{diag}(\sqrt{p})X\text{diag}(\sqrt{q})\|_{\text{tr}} \\ &= \min_{X=U'V} \frac{1}{2} \left(\sum_i p(i) \|U_i\|^2 + \sum_j q(j) \|V_j\|^2 \right) \end{aligned} \quad (12)$$

where $\text{diag}(\sqrt{p})$ is a diagonal matrix with $\sqrt{p(i)}$ on its diagonal (similarly $\text{diag}(\sqrt{q})$). The corresponding normalized complexity measure is given by $tc_{p,q}(X) = \|X\|_{\text{tr}(p,q)}^2$. Note that for a uniform distribution we have that $tc_{p,q}(X) = tc(X)$. Furthermore, it is easy to verify that for an “orthogonal” rank- k matrix X we have $tc_{p,q}(X) = k$ 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 $\|X^*\|_{\text{tr}(p,q)} = \sqrt{tc_{p,q}(X^*)} = \sqrt{k}$. How large a sample $S \cap A$ can we now shatter using such a weighted trace-norm? We can shatter a sample if $\|Y_{S \cap A}\|_{\text{tr}} \leq \sqrt{k}$. In order to calculate $\|Y_{S \cap A}\|_{\text{tr}}$, recall that for $(i, j) \in A$ we have $p(i) = q(j) = 1/(2n_A)$. We can now calculate: $\|Y_{S \cap A}\|_{\text{tr}(p,q)} = \left\| \sqrt{1/(2n_A)} Y_{S \cap A} \sqrt{1/(2n_A)} \right\|_{\text{tr}} = \|Y_{S \cap A}\|_{\text{tr}} / (2n_A) \leq$

$\sqrt{|S \cap A| n_A / (2n_A)} = \sqrt{|S| / (8n_A)}$. 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:

$$\min_X \|Y_S - X_S\|_F^2 + \lambda \|X\|_{\text{tr}(p,q)}, \quad (13)$$

as:

$$\begin{aligned} &\|Y_S - X_S\|_F^2 + \lambda \|X\|_{\text{tr}(p,q)} \\ &= (\|Y_{A \cap S} - X_{A \cap S}\|_F^2 + \lambda \|X_A\|_{\text{tr}(p,q)}) \\ &\quad + (\|Y_{B \cap S} - X_{B \cap S}\|_F^2 + \lambda \|X_B\|_{\text{tr}(p,q)}) \\ &= \left(\|Y_{A \cap S} - X_{A \cap S}\|_F^2 + \lambda/2 \sqrt{tc_A(X_A)} \right) \\ &\quad + \left(\|Y_{B \cap S} - X_{B \cap S}\|_F^2 + \lambda/2 \sqrt{tc_B(X_B)} \right) \end{aligned} \quad (14)$$

avoiding the scaling by the block sizes which we encountered in (11).

Returning to the synthetic experiments of Fig. 3, and comparing (11) with (14), 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 (14) is minimized (except for an overall scaling in λ). 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. Fig. 2, right panel, further shows the test error as a function on the constraint $tc_{p,q}(X)$.

Finally, Fig. 3 also suggests that the optimal shift is actually smaller than n_A/n_B . We consider a smaller

shift by using the partially-weighted trace-norm:

$$\begin{aligned} \|X\|_{\text{tr}(p,q,\alpha)} &= \left\| \text{diag}(p^{\alpha/2})X\text{diag}(q^{\alpha/2}) \right\|_{\text{tr}} \\ &= \min_{X=U^TV} \frac{1}{2} \left(\sum_i p(i)^\alpha \|U_i\|^2 + \sum_j q(j)^\alpha \|V_j\|^2 \right) \end{aligned} \quad (15)$$

And the corresponding normalized complexity measure $tc_{p,q,\alpha}(X) = \|X\|_{\text{tr}\left(\frac{p^\alpha}{n^{1-\alpha}}, \frac{q^\alpha}{m^{1-\alpha}}\right)}$.

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 (Salakhutdinov & Mnih, 2008; Koren, 2008).

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 (over the index pairs (i, j)) using partially-weighted trace-norm (Eq. 12) can be written as:

$$\begin{aligned} \sum_{\{i,j\} \in \mathcal{S}} \left((Y_{ij} - U_i^\top V_j)^2 + \right. & \quad (16) \\ \left. + \frac{\lambda}{2} \left(\frac{p(i)^\alpha}{n_i} \|U_i\|^2 + \frac{q(j)^\alpha}{m_j} \|V_j\|^2 \right) \right), \end{aligned}$$

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 (16) 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. $k > \min(n, m)$ (Burer & Monteiro, 2005). However, fitting orthogonal models in practice with 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 all of our experiments, we also replace unknown row $p(i)$ and column $q(j)$ marginals in (16) by their empirical estimates $\hat{p}(i) = n_i/|S|$ and $\hat{q}(j) = m_j/|S|$. This results in the following objective:

$$\begin{aligned} \sum_{\{i,j\} \in \mathcal{S}} \left((Y_{ij} - U_i^\top V_j)^2 + \right. & \quad (17) \\ \left. + \frac{\lambda}{2|S|} \left(n_i^{\alpha-1} \|U_i\|^2 + m_j^{\alpha-1} \|V_j\|^2 \right) \right). \end{aligned}$$

Table 1. Model performance using Root Mean Squared Error (RMSE) on the Netflix qualification set and the test set, that was randomly subsampled from the training data.

α	RMSE			RMSE		
	k	Test	Qual	k	Test	Qual
1	30	0.7607	0.9105	100	0.7412	0.9071
0.9	30	0.7573	0.9091	100	0.7389	0.9062
0.75	30	0.7723	0.9128	100	0.7491	0.9098
0.5	30	0.7823	0.9159	100	0.7613	0.9127
0	30	0.7889	0.9235	100	0.7667	0.9203

Setting $\alpha = 1$, corresponding to the weighted trace-norm (12), 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 are the actual SGD steps used by many practitioners on the Netflix dataset (Koren, 2008; Takács et al., 2009; Salakhutdinov & Mnih, 2008).

6. Experimental results

We evaluated various models on the Netflix dataset, which is the largest publicly available collaborative filtering dataset. The training set contains 100,480,507 ratings from 480,189 randomly-chosen, anonymous users on 17,770 movie titles. As part of the training data, Netflix also provides qualification set, containing 1,408,395 ratings. The pairs were selected from the most recent ratings for a subset of the users in the training dataset. Due to the special selection scheme, ratings from users with few ratings are overrepresented in the qualification set, relative to the training set. To avoid the issue of dealing with different training and test distributions, we also created our own validation and test sets, each containing 100,000 ratings that were randomly selected from the training set. As a baseline, Netflix provided the test score of its own system trained on the same data, which is 0.9514.

This dataset is interesting for several reasons. First, it is very large, and very sparse (98.8% sparse). Second, the dataset is very imbalanced with highly non-uniform samples. It includes users with over 10,000 ratings as well as users who rated fewer than 5 movies.

6.1. Results

In our first experiment, for various values of α , we fit parameters U and V using stochastic gradient descent as in (17) with $k = 30$. Both U and V were randomly initialized for all models and regularization parameters λ were chosen by cross-validation.

Performance results of the weighted trace-norm regu-

larization for various values of α are shown in table 1. Observe that the weighted trace-norm ($\alpha = 1$) achieved a RMSE of 0.9105 on the Netflix qualification set, significantly outperforming its unweighted counterpart with $\alpha = 0$, that achieved a RMSE of 0.9235. This large performance gap is striking. It clearly suggests that the weighting is quite important. Table 1 further reveals that the weighted trace-norm ($\alpha = 1$) is not optimal. Surprisingly, partially weighted trace-norm with $\alpha = 0.9$ achieved a RMSE of 0.9091, slightly outperforming the weighted matrix factorization. Performance results on the artificially created test set are similar to the results on the qualification set. Note also that the large gap in generalization performance between the test and the qualification sets is due to the Netflix’s special qualification selection scheme.

In our second experiment, we fitted much larger models with $k = 100$. As expected, the weighted trace-norm regularization ($\alpha = 1$) attained a RMSE 0.9071, significantly improving upon the unweighted model’s RMSE of 0.9203. Again, this large performance gap strongly suggests that the weighting can yield significant performance boost, particularly when dealing with very imbalanced data, such as the Netflix dataset.

In all of our experiments, we also empirically observed that for a wide range of regularization parameters λ , optimizing the weighted trace-norm almost always yielded better predictions on both the test and the Netflix qualification sets than optimizing the unweighted trace-norm. 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.

7. Discussion

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. Motivated by our analytic analysis, we further suggested a corrected version of the trace-norm, called weighted trace-norm, that does take into account the non-uniform sampling distribution. Our results on both synthetic and highly imbalanced Netflix datasets further demonstrate that the weighted trace-norm yields significant improvements in prediction quality. It is interesting to note that setting $\alpha = 1$ in the weighted trace-norm objective (12) implies that the frequent users (movies) get regularized much stronger than the rare users (movies). From Bayesian perspective, such regularization is quite unusual, since it effectively states that the effect of the prior becomes stronger as

we observe more data. Yet, our analysis and empirical results strongly suggest that in non-uniform setting, such “unorthodox” regularization is crucial for achieving good generalization performance.

Although theoretical guarantees are not the focus of this work, we hope that the weighted trace-norm, and the discussions in Sections 3 and 4, will be helpful in deriving theoretical learning guarantees for non-uniform sampling distributions, both in the form of generalization error bounds as in (Srebro & Shraibman, 2005), and generalizing the compressed-sensing inspired work on recovery of noisy low-rank matrices as in (Candes & Plan, 2009; Recht, 2009).

Acknowledgments

R.S. acknowledges the financial support from NSERC, Shell, and NTT Communication Sciences Laboratory.

References

- Abernethy, J., Bach, F., Evgeniou, T., and Vert, J.P. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research*, 10:803–826, 2009.
- Bach, F. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9:1019–1048, 2008.
- Burer, S. and Monteiro, R.D.C. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.
- Candes, E.J. and Plan, Y. Matrix completion with noise. *Proceedings of the IEEE (to appear)*, 2009.
- Candes, E.J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9, 2009.
- Candes, E.J. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inform. Theory (to appear)*, 2009.
- Fazel, M., Hindi, H., and Boyd, S.P. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings American Control Conference*, volume 6, 2001.
- Koren, Yehuda. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *ACM SIGKDD*, pp. 426–434, 2008.
- Recht, B. A simpler approach to matrix completion. preprint, available from author’s webpage, 2009.

- Rennie, J.D.M. and Srebro, N. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, pp. 719, 2005.
- Salakhutdinov, Ruslan and Mnih, Andriy. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.
- Srebro, N. and Shraibman, A. Rank, trace-norm and max-norm. In *COLT*, 2005.
- Srebro, N., Alon, N., and Jaakkola, T. Generalization error bounds for collaborative prediction with low-rank matrices. In *Advances In Neural Information Processing Systems 17*, 2005a.
- Srebro, N., Rennie, J., and Jaakkola, T. Maximum margin matrix factorization. In *Advances In Neural Information Processing Systems 17*, 2005b.
- Takács, Gábor, Pilászy, István, Németh, Bottyán, and Tikk, Domonkos. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656, 2009.