Local Low-Rank Matrix Approximation

Joonseok Lee JLEE716@GATECH.EDU
Seungyeon Kim SEUNGYEON.KIM@GATECH.EDU
Guy Lebanon LEBANON@CC.GATECH.EDU
Georgia Institute of Technology, 266 Ferst Dr., Atlanta, GA 30363 USA

Yoram Singer SINGER@GOOGLE.COM
Google Research, 1600 Amphitheatre Parkway Mountain View, CA 94043 USA

Abstract

Matrix approximation is a common tool in recommendation systems, text mining, and computer vision. A prevalent assumption in constructing matrix approximations is that the partially observed matrix is of low-rank. We propose a new matrix approximation model where we assume instead that the matrix is *locally* of low-rank, leading to a representation of the observed matrix as a weighted sum of low-rank matrices. We analyze the accuracy of the proposed local low-rank modeling. Our experiments show improvements in prediction accuracy over classical approaches for recommendation tasks.

1. Introduction

Matrix approximation is a common task in machine learning. Given a few observed matrix entries $\{M_{a_1,b_1}, \ldots, M_{a_m,b_m}\}$, matrix completion constructs a matrix $\hat{M}$ that approximates M at its unobserved entries. Matrix approximation is used heavily in recommendation systems, text processing, computer vision, and bioinformatics. In recommendation systems, for example, the matrix M corresponds to ratings of items (columns) by users (rows). Matrix approximation in this case corresponds to predicting the ratings of all users on all items based on a few observed ratings. In many cases, matrix approximation leads to state-of-the-art models that are used in industrial settings.

In general, the problem of completing a matrix M based on a few observed entries is ill-posed. There are infinite number of matrices that perfectly agree with the observed entries of M, so without additional assumptions, it is hard to prefer some matrices over others as candidates for $\hat{M}$. One popular assumption is that M is a low-rank matrix, which suggests that it is reasonable to assume that the completed matrix $\hat{M}$ has low-rank. More formally, we approximate a matrix $M \in \mathbb{R}^{n_1 \times n_2}$ by a rank r matrix $\hat{M} = UV^T$, where $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{n_2 \times r}$, and $r \ll \min(n_1, n_2)$. In many real datasets, the low-rank assumption is realistic. Further, low-rank approximations often yield matrices that generalizes well to the unobserved entries.

In this paper, we extend low-rank matrix approximation in a way that significantly relaxes the low-rank assumption. Instead of assuming that M has low-rank globally, we assume that M behaves as a low-rank matrix in the vicinity of certain row-column combinations. We therefore construct several low-rank approximations of M, each being accurate in a particular region of the matrix. We express our estimator as a smoothed convex combination of low-rank matrices each of which approximates M in a local region.

We use techniques from non-parametric kernel smoothing to achieve two goals. The first goal is develop a notion of local low-rank approximation, and the second is the aggregation of several local models into unified matrix approximation. Standard low-rank matrix approximation techniques achieve consistency in the limit of large data (convergence to the data generating process) assuming that M is low-rank. Our local method achieves consistency without the low-rank assumption. Instead, we require that sufficient samples are available in increasingly small neighborhoods. This analysis mirrors the theory of non-parametric kernel smoothing, that is primarily developed for continuous spaces, and generalizes well-known compressed

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

sensing results to our setting. Our experiments show that local low-rank modeling is significantly more accurate than global low-rank modeling in the context of recommendation systems.

2. Low-rank matrix approximation

We describe in this section two standard approaches for low-rank matrix approximation (LRMA). We start by establishing the notation used throughout the paper. We denote matrices using upper case letters. The original (partially observed) matrix is denoted by $M \in \mathbb{R}^{n_1 \times n_2}$. A low-rank approximation of M is denoted by $\hat{M} = UV^T$, where $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{n_2 \times r}$, and $r \ll \min(n_1, n_2)$. The set of integers $\{1, \ldots, n\}$ is abbreviated as $[n]$. The set of observed entries of M is denoted by $\mathtt{A} \stackrel{\text{def}}{=} \{(a_1, b_1), \ldots, (a_m, b_m)\} \subseteq [n_1] \times [n_2]$. The training set is therefore $\{M_{a,b} : (a, b) \in \mathtt{A}\}$. Mappings from matrix indices to a matrix space are denoted in calligraphic letters, e.g. $\mathcal{T}$, and are operators of the form $\mathcal{T} : [n_1] \times [n_2] \to \mathbb{R}^{n_1 \times n_2}$. We denote the entry (i, j) of the matrix $\mathcal{T}(a, b)$ as $\mathcal{T}_{i,j}(a, b)$. A projection $\Pi_{\mathtt{A}}$ with respect to a set of matrix indices $\mathtt{A}$ is the function $\Pi_{\mathtt{A}} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{n_1 \times n_2}$ defined by

$$[\Pi_{\mathtt{A}}(M)]_{a,b} \stackrel{\text{def}}{=} \begin{cases} M_{a,b} & (a, b) \in \mathtt{A} \\ 0 & \text{otherwise.} \end{cases}$$

We denote by $\odot$ the component-wise product of two matrices $[A \odot B]_{i,j} = A_{i,j} B_{i,j}$. We use in this paper three matrix norms: the Frobenius norm $\|X\|_F \stackrel{\text{def}}{=} \sqrt{\sum_i \sum_j X_{i,j}^2}$, the sup-norm $\|X\|_\infty \stackrel{\text{def}}{=} \sup_{i,j} |X_{i,j}|$, and the nuclear norm $\|X\|_* \stackrel{\text{def}}{=} \sum_{i=1}^{r} \sigma_i(X)$, where $\sigma_i(X)$ is the i'th singular value of X (for symmetric matrices $\|X\|_* = \text{trace}(X)$).

Below are two popular approaches for constructing a low-rank approximation $\hat{M}$ of M. The first is based on minimizing the Frobenius norm of $\Pi_{\mathtt{A}}(M - \hat{M})$ and the second is based on minimizing the nuclear norm of a matrix satisfying constraints constructed from the training set.

A1: Incomplete SVD. The incomplete SVD method constructs a low-rank approximation $\hat{M} = UV^T$ by solving

$$(U, V) = \arg\min_{U,V} \sum_{(a,b) \in \mathtt{A}} ([UV^T]_{a,b} - M_{a,b})^2 \ , \qquad (1)$$

or equivalently

$$\hat{M} = \arg\min_{X} \|\Pi_{\mathtt{A}}(X - M)\|_F \ \text{ s.t. } \text{rank}(X) = r. \quad (2)$$

A2: Compressed Sensing. An alternative to (2) that originated from the compressed sensing community (Candès & Tao, 2010) is to minimize the nuclear norm of a matrix subject to constraints constructed from the observed entries:

$$\hat{M} = \arg\min_{X} \|X\|_* \quad \text{s.t.} \quad \|\Pi_{\mathtt{A}}(X - M)\|_F < \epsilon \ . \quad (3)$$

Minimizing the nuclear norm $\|X\|_*$ is an effective surrogate for minimizing the rank of X, and solving (3) results in a low-rank matrix $\hat{M} = UV^T$ that approximates the matrix M. One advantage of *A2* over *A1* is that we do not need to constrain the rank of $\hat{M}$ in advance. Note also that the problem defined by (3), while being convex, may not necessarily scale up easily to large matrices.

3. Local low-rank matrix approximation

In order to facilitate a local low-rank matrix approximation, we need to pose an assumption that there exists a metric structure over $[n_1] \times [n_2]$. The distance $d((a, b), (a', b'))$ reflects the similarity between the rows a and a' and columns b and b'. In the case of recommendation systems, for example, $d((a, b), (a', b'))$ expresses the relationship between users a, a' and items b, b'. The distance function may be constructed using the observed ratings $\Pi_{\mathtt{A}}(M)$ or additional information such as item-item similarity or side information on the users when available. See Section 7 for further details.

In the global matrix factorization setting in Section 2, we assume that the matrix $M \in \mathbb{R}^{n_1 \times n_2}$ has a low-rank structure. In the local setting, however, we assume that the model is characterized by multiple low-rank $n_1 \times n_2$ matrices. Specifically, we assume a mapping $\mathcal{T} : [n_1] \times [n_2] \to \mathbb{R}^{n_1 \times n_2}$ that associates with each row-column combination $[n_1] \times [n_2]$ a low rank matrix that describes the entries of M in its neighborhood (in particular this applies to the observed entries $\mathtt{A}$):

$$\mathcal{T} : [n_1] \times [n_2] \to \mathbb{R}^{n_1 \times n_2} \quad \text{where} \quad \mathcal{T}_{a,b}(a, b) = M_{a,b} \ .$$

Figures 1 and 2 illustrate this model[1].

Without additional assumptions, it is impossible to estimate the mapping $\mathcal{T}$ from a set of $m < n_1 n_2$ observations. We assume, as is often done in non-parametric statistics, that the mapping $\mathcal{T}$ is slowly varying. See the formal definition in the sequel and an illustration in Figure 2. Since the domain of $\mathcal{T}$ is discrete, the classical definitions of continuity or differentiability are not

[1] For illustrative purposes, we assume in Figure 1 a distance function d whose neighborhood structure coincides with the natural order on indices. That is, $s = (a, b)$ is similar to $r = (c, d)$ if $|a - c|$ and $|b - d|$ are small.

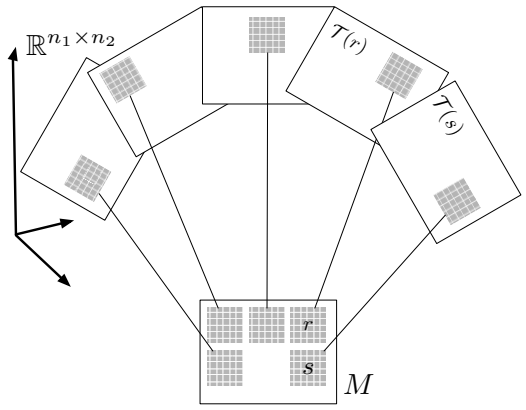

Figure 1. For all $s \in [n_1] \times [n_2]$, the neighborhood $\{s' : d(s, s') < h\}$ in the original matrix M is approximately described by the corresponding entries of the low-rank matrix $\mathcal{T}(s)$ (shaded regions of M are matched by lines to the corresponding regions in $\mathcal{T}(s)$ that approximate them). If $d(s, r)$ is small (see footnote 1), $\mathcal{T}(s)$ is similar to $\mathcal{T}(r)$, as shown by their spatial closeness in the embedding $\mathbb{R}^{n_1 \times n_2}$.

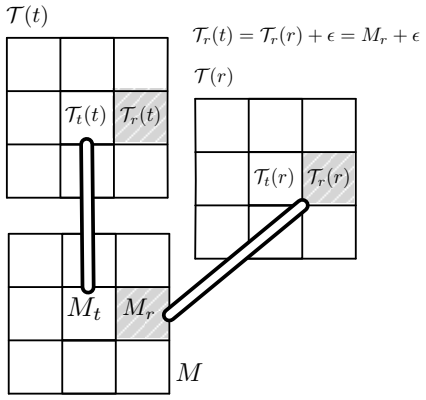

Figure 2. The original matrix M (bottom) is described locally by two low-rank matrices $\mathcal{T}(t)$ (near t) and $\mathcal{T}(r)$ (near r). The lines connecting the three matrices identify identical entries: $M_t = \mathcal{T}_t(t)$ and $M_r = \mathcal{T}_r(r)$. The equation at the top right shows a relation tying the three patterned entries. Assuming the distance $d(t, r)$ is small, $\epsilon = \mathcal{T}_r(t) - \mathcal{T}_r(r) = \mathcal{T}_r(t) - M_r(r)$ is small as well.

applicable in our setting. We assume instead that $\mathcal{T}$ is Hölder continuous (see Definition 1 in Section 5).

Following common approaches in non-parametric statistics, we define a smoothing kernel $K_h(s_1, s_2)$, $s_1, s_2 \in [n_1] \times [n_2]$, as a non-negative symmetric unimodal function that is parameterized by a bandwidth parameter $h > 0$. A large value of h implies that $K_h(s, \cdot)$ has a wide spread, while a small h corresponds to narrow spread of $K_h(s, \cdot)$. Three popular smoothing kernels are the uniform kernel, the triangular kernel, and the Epanechnikov kernel, defined respectively as

$$K_h(s_1, s_2) \propto \mathbf{1}[d(s_1, s_2) < h] \tag{4}$$

$$K_h(s_1, s_2) \propto (1 - h^{-1} d(s_1, s_2))\, \mathbf{1}[d(s_1, s_2) < h] \tag{5}$$

$$K_h(s_1, s_2) \propto (1 - d(s_1, s_2)^2)\, \mathbf{1}[d(s_1, s_2) < h] \ . \tag{6}$$

We denote by $K_h^{(a,b)}$ the matrix whose (i, j)-entry is $K_h((a, b), (i, j))$. See for instance (Wand & Jones, 1995) for more information on smoothing kernels.

We describe below the local extensions of incomplete SVD (*A1*) and compressed sensing (*A2*) matrix approximations. Both extensions estimate $\mathcal{T}(a, b)$ in the vicinity of $(a, b) \in [n_1] \times [n_2]$ given the samples $\Pi_{\mathtt{A}}(M)$.

Local-A1: Incomplete SVD

$$\hat{\mathcal{T}}(a, b) = \arg\min_X \|K_h^{(a,b)} \odot \Pi_{\mathtt{A}}(X - M)\|_F \tag{7}$$
$$\text{s.t.} \quad \text{rank}(X) = r \ .$$

Local-A2: Compressed Sensing

$$\hat{\mathcal{T}}(a, b) = \arg\min_X \|X\|_* \tag{8}$$
$$\text{s.t.} \quad \|K_h^{(a,b)} \odot \Pi_{\mathtt{A}}(X - M)\|_F < \epsilon \ .$$

The two optimization problems above describe how to estimate $\hat{\mathcal{T}}(a, b)$ for a particular choice of $(a, b) \in [n_1] \times [n_2]$. Conceptually, this technique can be applied at each test entry (a, b), resulting in the matrix approximation $\hat{M} \approx M$ where

$$\hat{M}_{a,b} = \hat{\mathcal{T}}_{a,b}(a, b), \quad (a, b) \in [n_1] \times [n_2] \ .$$

However, such a construction would require solving a non-linear optimization problem for each test index (a, b) and is thus computationally prohibitive. Instead, we describe in the next section how to use a set of q local models $\hat{\mathcal{T}}(s_1), \ldots, \hat{\mathcal{T}}(s_q)$, $s_1, \ldots, s_q \in [n_1] \times [n_2]$ to obtain a computationally efficient estimate $\hat{\hat{\mathcal{T}}}(s)$ for all $s \in [n_1] \times [n_2]$.

4. Global Approximation

The problem of recovering a mapping $\mathcal{T}$ from q values without imposing a strong parametric form is known as non-parametric regression. We propose using a variation of locally constant kernel regression (Wand & Jones, 1995), also known as Nadaraya-Watson regression

$$\hat{\hat{\mathcal{T}}}(s) = \sum_{i=1}^{q} \frac{K_h(s_i, s)}{\sum_{j=1}^{q} K_h(s_j, s)} \hat{\mathcal{T}}(s_i) \ . \tag{9}$$

Equation (9) is simply a weighted average of $\hat{\mathcal{T}}(s_1), \ldots, \hat{\mathcal{T}}(s_q)$, where the weights ensure that values of $\hat{\mathcal{T}}$ at indices close to s contribute more than those further away from s. Note that both the left-hand side and the right-hand side of (9) denote matrices. The denominator in (9) ensures that the weights sum to one.

In contrast to $\hat{\mathcal{T}}$, the estimate $\hat{\hat{\mathcal{T}}}$ can be computed for all $s \in [n_1] \times [n_2]$ efficiently since computing $\hat{\hat{\mathcal{T}}}(s)$ simply requires evaluating and averaging $\hat{\mathcal{T}}(s_i)$, $i = 1, \ldots, q$. The resulting matrix approximation is $\hat{\hat{M}}_{a,b} = \hat{\hat{\mathcal{T}}}_{a,b}(a, b)$ and $(a, b) \in [n_1] \times [n_2]$.

The accuracy of $\hat{\hat{\mathcal{T}}}$ as an estimator of $\hat{\mathcal{T}}$ improves with the number of local models q and the degree of continuity of $\hat{\mathcal{T}}$. The accuracy of $\hat{\hat{\mathcal{T}}}$ as an estimator of $\mathcal{T}$ is limited by the quality of the local estimators $\hat{\mathcal{T}}(s_1), \ldots, \hat{\mathcal{T}}(s_q)$. However, assuming that $\hat{\mathcal{T}}(s_1), \ldots, \hat{\mathcal{T}}(s_q)$ are accurate in the neighborhoods of $s_1, \ldots, s_q$, and q is sufficiently large, the estimation error $\hat{\hat{\mathcal{T}}}_{a,b}(a, b) - \mathcal{T}_{a,b}(a, b)$ is likely to be small as we analyze in the next section. We term the resulting approach LLORMA for Local LOw Rank Matrix Approximation.

5. Estimation accuracy

In this section we analyze the estimation accuracy of LLORMA. Our analysis consists of two parts. In the first we analyze the large deviation of $\hat{\mathcal{T}}$ from $\mathcal{T}$. Then, based on this analysis, we derive a deviation bound on the global approximation $\hat{\hat{\mathcal{T}}}$. Our analysis technique is based on the seminal paper of Candès & Tao (2010). The goal of this section is to underscore the characteristics of estimation error in terms of parameters such as the train set size, matrix dimensions, and kernel bandwidth.

Analysis of $\hat{\mathcal{T}} - \mathcal{T}$

Candès & Tao (2010) established that it is possible to estimate an $n_1 \times n_2$ matrix M of rank r if the number of observations $m \geq C\mu r n \log^6 n$, where $n = \min(n_1, n_2)$, C is a constant, and μ is the strong incoherence property parameter described in (Candès & Tao, 2010). This bound is tight in the sense that it is close to the information theoretic limit of $\Omega(r\, n \log n)$.

The aforementioned result is not applicable in our case since the matrix M is not necessarily of low-rank. Concretely, when $r = O(n)$ the bound above degenerates into a sample complexity of $O(n^2 \log n)$ which is clearly larger than the number of entries in the matrix M. We develop below a variation on the results in (Candès & Tao, 2010) and (Candès & Plan, 2010) that applies to the *local-A2* compressed-sensing estimator $\hat{\mathcal{T}}$.

Definition 1. Let X be a metric space. A function $f : X \to \mathbb{R}^{n_1 \times n_2}$ is *Hölder continuous* with parameters $\alpha, \beta > 0$ if

$$\forall x, x' \in X : \ \|f(x) - f(x')\|_F \leq \alpha\, d^\beta(x, x') \ . \quad (10)$$

In our analysis we make the following assumptions: (i) $\mathcal{T}$ is Hölder continuous, (ii) $\mathcal{T}(s)$ is a rank r matrix that satisfies the strong incoherence property, and (iii) the kernel K_h is a uniform kernel based on a product distance function. The Hölder continuity assumption on $\mathcal{T}$ can be replaced by the following weaker condition without affecting the results

$$\|K_h^s \odot (\mathcal{T}(s) - \mathcal{T}(s'))\|_F \leq \alpha d^\beta(s, s'). \quad (11)$$

We denote by $B_h(s)$ the neighborhood of indices near s, $B_h(s) \stackrel{\text{def}}{=} \{s' \in [n_1] \times [n_2] : d(s, s') < h\}$ and we use $n_1(h, s)$ and $n_2(h, s)$ to denote the number of unique row and column indices, respectively, in $B_h(s)$. Finally, we denote $\gamma = \min(n_1(h, s), n_2(h, s))$.

The proposition below provides a bound on the average squared-error within a neighborhood of s

$$\mathcal{E}(\hat{\mathcal{T}})(s, h) = \sqrt{\frac{1}{|B_h(s)|} \sum_{s' \in B_h(s)} \left(\hat{\mathcal{T}}_{s'}(s) - \mathcal{T}_{s'}(s)\right)^2} \ .$$

Proposition 1. *If* $|\mathtt{A} \cap B_h(s)| \geq C\mu^2 \gamma r \log^6 \gamma$, *then with probability of at least* $1 - \delta$,

$$\mathcal{E}(\hat{\mathcal{T}})(s, h) \leq \frac{\alpha h^\beta}{\sqrt{|B_h(s)|}} \left(4\sqrt{\frac{\gamma(2 + p)}{p}} + 2\right) \ ,$$

where $\gamma = \sqrt[3]{1/\delta}$ *and* $p = |\mathtt{A} \cap B_h(s)|/|B_h(s)|$.

Proof. Assumptions (i) and (iii) above imply that if $K_h(s, s') > 0$ then

$$\|K_h^s \odot (\mathcal{T}(s) - \mathcal{T}(s'))\|_\infty < \alpha h^\beta \ .$$

We can thus assume that if $d(s, s') < h$, an observation $M_{s'} = \mathcal{T}_{s'}(s')$ is equal to $\mathcal{T}_{s'}(s) + Z$ where Z is a random variable whose absolute value is bounded by αh^β. This means that we can use observations $M_{s'} = \mathcal{T}_{s'}(s')$ for estimating the local model $\mathcal{T}(s)$ as long as we admit a noisy measurement process.

Since K is a uniform kernel based on a product distance by assumption (iii), the set $B_h(s)$ is a Cartesian product set. We view this product set as a matrix of dimensions $n_1(h, s) \times n_2(h, s)$ that we approximate. (Note that $n_1(h, s)$ and $n_2(h, s)$ are monotonically increasing with h, and as $h \to \infty$, $n_1(h, s) = n_1$,

$n_2(h,s) = n_2$.) The number of observed entries in this matrix approximation problem is $|\mathtt{A} \cap B_h(s)|$.

Applying Theorem 7 in (Candès & Plan, 2010) to the matrix completion problem described above, we get that if $|\mathtt{A} \cap B_h(s)| \geq C\mu^2 \gamma r \log^6 \gamma$, then with probability greater than $1 - \gamma^{-3}$,

$$\|K_h^s \odot (\mathcal{T}(s) - \hat{\mathcal{T}}(s))\|_F \leq \alpha h^\beta \left(4\sqrt{\frac{\gamma(2+p)}{p}} + 2 \right),$$

where $p = \frac{|\mathtt{A} \cap B_h(s)|}{|B_h(s)|}$ is the density of observed samples. Dividing by $\sqrt{|B_h(s)|}$ concludes the proof. □

If the observed samples are spread uniformly over the matrix, we have $p = m/(n_1 n_2)$, so

$$\begin{aligned} 4\sqrt{\gamma\frac{2+p}{p}} + 2 &= 4\sqrt{\gamma\frac{2 + m/(n_1 n_2)}{m/(n_1 n_2)}} + 2 \\ &= 4\sqrt{\frac{\gamma(2n_1 n_2 + m)}{m}} + 2. \end{aligned}$$

Multiplying $\alpha h^\beta / \sqrt{|B_h(s)|}$ yields Corollary 1.

Corollary 1. Assume that the conditions of Proposition 1 hold and in addition the observed samples are spread uniformly with respect to d. Then, the following inequality holds

$$\mathcal{E}(\hat{\mathcal{T}})(s,h) \leq \frac{4\alpha h^\beta}{\sqrt{|B_h(s)|}} \sqrt{\frac{\gamma(2n_1 n_2 + m)}{m}} + \frac{2\alpha h^\beta}{\sqrt{|B_h(s)|}}.$$

If in addition the matrix M is squared ($n_1 = n_2 = n$) and the distribution of distances d is uniform, then $n_1(h,s) = n_2(h,s) = n/h$, $|B_h(s)| = (n/h)^2$, and $\gamma = n/h$. In this case, the bound on $\mathcal{E}(\hat{\mathcal{T}})(s,h)$ becomes

$$4\alpha h^{\beta+1/2} \sqrt{\frac{2n}{m} + \frac{1}{n}} + \frac{2\alpha h^{\beta+1}}{n}. \tag{12}$$

In the case of a square matrix with uniformly spread samples, it is instructive to view n, m, h as monotonically increasing sequences, indexed by $k \in \mathbb{N}$ and assume that $\lim_{k\to\infty} n_{[k]} = \lim_{k\to\infty} m_{[k]} = \infty$. In other words, we consider the limit of matrices of increasing sizes with an increasing number of samples. In the case of uniformly distributed distances, the bound (12) will converge to zero if

$$\lim_{k\to\infty} \frac{h_{[k]}^{\beta+1}}{n_{[k]}} = \lim_{k\to\infty} \frac{h_{[k]}^{2\beta+1}}{n_{[k]}} = \lim_{k\to\infty} \frac{h_{[k]}^{2\beta+1} n_{[k]}}{m_{[k]}} = 0.$$

Analysis of $\hat{\hat{\mathcal{T}}} - \mathcal{T}$

We start by showing that $\hat{\mathcal{T}}$ is Hölder continuous with high probability, and then proceed to analyze the estimation error of $\hat{\hat{\mathcal{T}}}$.

Proposition 2. *If $d(s,s') < h$ and Proposition 1 holds at s, s', then with probability at least $1 - \delta$,*

$$\|K_h^s \odot (\hat{\mathcal{T}}(s) - \hat{\mathcal{T}}(s'))\|_F \leq \alpha h^\beta \left(8\sqrt{\frac{\gamma(2+p)}{p}} + 5 \right).$$

where $\gamma = \sqrt[3]{2/\delta}$.

Proof. Using the triangle inequality for $\|\cdot\|_F$,

$$\begin{aligned} \|K_h^s \odot (\hat{\mathcal{T}}(s) - \hat{\mathcal{T}}(s'))\|_F \quad \leq \quad & \|K_h^s \odot (\hat{\mathcal{T}}(s) - \mathcal{T}(s))\|_F \\ + \quad & \|K_h^s \odot (\hat{\mathcal{T}}(s') - \mathcal{T}(s'))\|_F \\ + \quad & \|K_h^s \odot (\mathcal{T}(s) - \mathcal{T}(s'))\|_F. \end{aligned}$$

We apply the bound from Proposition 1 to the first two terms and use the assumption that $\mathcal{T}$ is Hölder continuous to bound the third term. The adjustment to the confidence level $2\gamma^{-3}$ is obtained using the union bound. □

Proposition 3. *Assume that Proposition 1 holds. Then, with probability of at least $1 - \delta$,*

$$\mathcal{E}(\hat{\hat{\mathcal{T}}})(s,h) \leq \frac{\alpha h^\beta}{\sqrt{|B_h(s)|}} \left(12\sqrt{\frac{\gamma(2+p)}{p}} + 7 \right).$$

where $\gamma = \sqrt[3]{(2|\mathtt{A} \cap B_h(s)| + 1)/\delta}$.

Proof. Using the triangle inequality we get

$$\begin{aligned} &\|K_h^s \odot (\hat{\hat{\mathcal{T}}}(s) - \mathcal{T}(s))\|_F \leq \\ &\quad \|K_h^s \odot (\hat{\mathcal{T}}(s) - \mathcal{T}(s))\|_F + \|K_h^s \odot (\hat{\hat{\mathcal{T}}}(s) - \hat{\mathcal{T}}(s))\|_F. \end{aligned} \tag{13}$$

We bound the first term using Proposition 1. Since $\hat{\hat{\mathcal{T}}}(s)$ is a weighted average of $\hat{\mathcal{T}}(s_i)$, $i = 1, \ldots, q$ with $s_i \in B_h(s)$, the second term is bounded by

$$\begin{aligned} &\|K_h^s \odot (\hat{\hat{\mathcal{T}}}(s) - \hat{\mathcal{T}}(s))\|_F \\ &\quad = \left\| K_h^s \odot \left(\sum_i \frac{w_i}{\sum_j w_j} \hat{\mathcal{T}}(s_i) - \hat{\mathcal{T}}(s) \right) \right\|_F \\ &\quad = \left\| K_h^s \odot \sum_i \frac{w_i}{\sum_j w_j} (\hat{\mathcal{T}}(s_i) - \hat{\mathcal{T}}(s)) \right\|_F \\ &\quad \leq \sum_i \left\| \frac{w_i}{\sum_j w_j} K_h^s \odot (\hat{\mathcal{T}}(s_i) - \hat{\mathcal{T}}(s)) \right\|_F \\ &\quad \leq \sum_i \frac{w_i}{\sum_j w_j} \|K_h^s \odot (\hat{\mathcal{T}}(s_i) - \hat{\mathcal{T}}(s))\|_F. \end{aligned}$$

There are $|\mathtt{A} \cap B_h(s)|$ summands in the above term. We bound each of them using Proposition 2. Together with the bound (13) this gives the desired result (after dividing by $\sqrt{|B_h(s)|}$). The adjustment to the confidence level $(2|\mathtt{A} \cap B_h(s)| + 1)\gamma^{-3}$ is obtained using the union bound. □

The constants in the proposition above can be improved considerably by using large deviation bounds that are tighter than the union bound.

6. The LLORMA algorithm

In the previous sections, we assumed a general kernel function $K_h(s_1, s_2)$, where $s_1, s_2 \in [n_1] \times [n_2]$. This kernel function may be defined in several ways. For simplicity, we assume a product form $K_h((a, b), (c, d)) = K_{h_1}(a, c)K'_{h_2}(b, d)$ where K and K' are kernels on the spaces $[n_1]$ and $[n_2]$, respectively. We used the Epanechnikov kernel (6) for both K, K' as it achieves the lowest integrated squared error (Wand & Jones, 1995).

The distance d in (6) may be defined using additional information describing row (user) similarity or column (item) similarity. If no such information is available (as is the case in our experiments), d may be computed solely based on the partially observed matrix M. In that case, we may use any distance measure between two row vectors (for K) or two column vectors (for K'). Empirically, we found that standard distance measures such as L_2 or cosine similarity do not perform well when M is sparse. We therefore instead factorize M using standard incomplete SVD (1) $M \approx UV^T$ and then proceed to compute d based on the cosine distances between the rows of factor matrices U and V. For example, the distance between users i and j is $d(i, j) = \arccos\left(\frac{\langle u_i, u_j \rangle}{\|u_i\| \cdot \|u_j\|}\right)$, where u_i, u_j are the i and j rows of the matrix U.

There are several ways of choosing the anchor points $s_1, \ldots, s_q$ that define the estimator $\hat{\hat{\mathcal{T}}}$.

1. Sample anchor points uniformly from $[n_1] \times [n_2]$.
2. Sample anchor points uniformly from the observed entries (training set) $\mathtt{A}$.
3. Sample anchor points from the test entries (if they are known in advance).
4. Select anchor points such that no entry in $[n_1] \times [n_2]$ is far away from an anchor point.

If the row and column indices corresponding to the test set are known in advance, method 3 above is preferred. We did not find a significant difference between methods 1 and 2 empirically, so we used method 2 in our experiments.

Algorithm 1 The LLORMA Algorithm

1: **Input:** $M \in \mathbb{R}^{n_1 \times n_2}, h_1, h_2, r, q$
2: **for all** $t = 1, ..., q$ **in parallel do**
3: $(a_t, b_t) :=$ a randomly selected train element
4: **for** $i = 1 \to n_1$ **do**
5: $[K_{h_1}^{(a_t)}]_i := (1 - d_a(a_t, i)^2)\mathbf{1}_{\{d_a(a_t,i)<h\}}$
6: **end for**
7: **for** $j = 1 \to n_2$ **do**
8: $[K_{h_2}^{(b_t)}]_j := (1 - d_b(b_t, j)^2)\mathbf{1}_{\{d_b(b_t,j)<h\}}$
9: **end for**
10: $(U^{(t)}, V^{(t)}) := \arg\min_{U,V}[\lambda_U \Omega(U) + \lambda_V \Omega(V)$
11: $+ \sum_{(i,j)\in\mathtt{A}} [K_{h_1}^{(a_t)}]_i [K_{h_2}^{(b_t)}]_j \left([UV^T]_{i,j} - M_{i,j}\right)^2]$
12: **end for**
13: **Output:** $\hat{\mathcal{T}}(s_t) = U^{(t)}V^{(t)^T}, t = 1, \ldots, q$

Algorithm 1 describes the learning algorithm for estimating the local models at the anchor points $\hat{\mathcal{T}}(s_i)$, with $i = 1, \ldots, q$. In line 10, we use L_2 regularization, as is standard in global SVD. This minimization problem can be computed with gradient-based methods, as it is differentiable. After these models are estimated, they are combined using (9) to create the estimate $\hat{\hat{\mathcal{T}}}(s)$ for all $s \in [n_1] \times [n_2]$.

The q iterations of the loop in Algorithm 1 (lines 2-12) are independent of each other, and thus can be computed in parallel. The complexity of Algorithm 1 is q times the complexity of solving a single regularized SVD problem. Note, however, that Algorithm 1 may be in fact faster than global SVD since (a) the q loops may be computed in parallel, and (b) the rank used in the local SVD model can be significantly lower than the rank used in a global SVD model (see Section 7). If the kernel K_h has limited support ($K_h(s, s')$ is non-zero only for a few values of s' for any given s) the regularized SVD problems in Algorithm 1 would be more sparse than the global SVD problem, resulting in an additional speedup.

7. Experiments

Experimental Design. We conduct two experiments using recommendation systems data: (a) comparing LLORMA to SVD and other state-of-the-art techniques, and (b) examining dependency of LLORMA on the rank r, the number of anchor points q, and the train set size.

We use three popular recommendation systems datasets: MovieLens 1M ($6K \times 4K$ with 10^6 observations), MovieLens 10M ($70K \times 10K$ with 10^7 obser-

Method	MovieLens (1M)		MovieLens (10M)		Netflix	
APG	Not known		0.8005		0.8433	
DFC(NYS)	Not known		0.8085		0.8486	
DFC(PROJ)	Not known		0.7944		0.8411	
LRMA	Global	Local	Global	Local	Global	Local
Rank-1	0.9201	0.9135	0.8723	0.8650	0.9388	0.9295
Rank-3	0.8838	0.8670	0.8348	0.8189	0.8928	0.8792
Rank-5	0.8737	0.8537	0.8255	0.8049	0.8836	0.8604
Rank-7	0.8678	0.8463	0.8234	0.7950	0.8788	0.8541
Rank-10	0.8650	0.8396	0.8219	0.7889	0.8765	0.8444
Rank-15	0.8652	0.8370	0.8225	0.7830	0.8758	0.8365
Rank-20	0.8647	0.8333	0.8220	0.7815	0.8742	0.8337

Table 1. RMSE of different recommendation systems on three datasets: MovieLens 1M, MovieLens 10M, and Netflix. Results on APG (Toh & Yun, 2010) and DFC are from (Mackey et al., 2011).

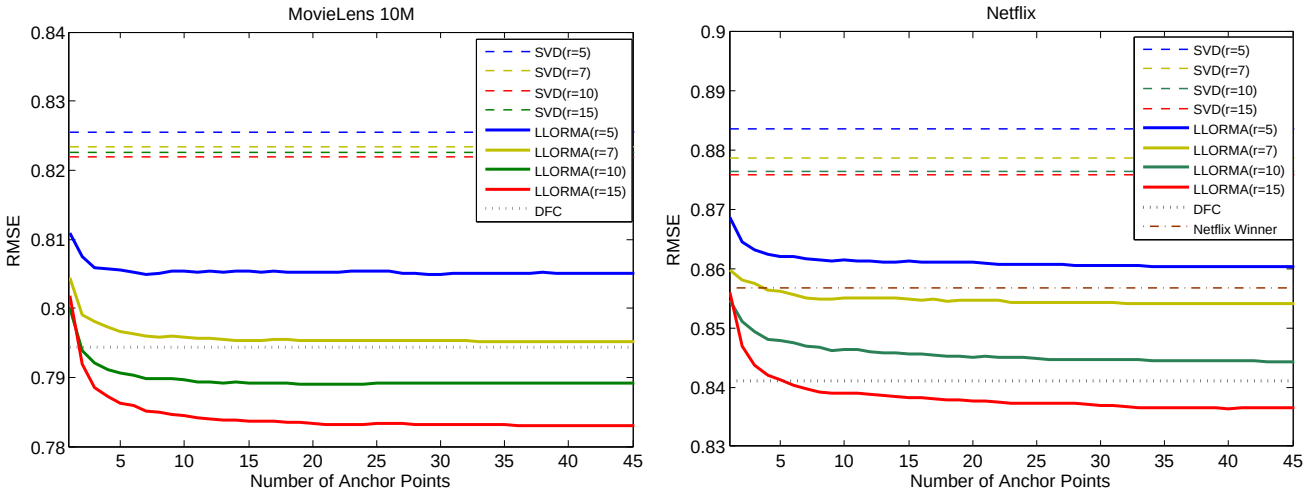

Figure 3. RMSE of LLORMA, SVD, and other baselines on MovieLens 10M (left) and Netflix (right) dataset. (The Netflix winner RMSE is based on the qualifying set from the Netflix competition while our result use a randomly sampled dataset of similar size.) LLORMA models are indicated by thick solid lines, while SVD models are indicated by dotted lines. Models with same rank are colored identically.

vations), and Netflix ($480K \times 18K$ with 10^8 observations). We split the available data to train and test sets randomly such that the ratio of train set to test set is 9:1, and averaged over five such repetitions. We use a default rating of 3 for test users or items without training observations.

In our experiments, we used the Epanechnikov kernel with $h_1 = h_2 = 0.8$, $\mu = 0.01$ (gradient descent step-size), $\lambda_U = \lambda_V = 0.001$ (L_2-regularization coefficient), $T = 100$ (maximum number of iterations), and $\epsilon = 0.0001$ (gradient descent convergence threshold), and $q = 50$ (number of anchor points).

Result and Analysis. Table 1 lists the performance of LLORMA with 50 anchor points, SVD, and recent state-of-the-art methods based on figures published in (Mackey et al., 2011). For a fixed rank r, LLORMA always outperforms SVD. Both LLORMA and SVD perform better as r increases. Although both encounter the law of diminishing returns, LLORMA with a modest rank of $r \geq 5$ outperforms SVD with any rank whatsoever. We can see that LLORMA also outperforms the Netflix winner RMSE 0.8567 as well as other baselines.

Figure 3 graphs the RMSE of LLORMA and SVD (for several values of r) as well as the recently proposed DFC method (Mackey et al., 2011) as a function of the number of anchor points (all but LLORMA are constants in this variable). As in the case of Table 1, both LLORMA and SVD improve as r increases, but LLORMA with rank $r \geq 5$ outperforms SVD with any rank. Moreover, LLORMA outperforms SVD in average with even a few anchor points (though the performance of LLORMA improves further as the number of

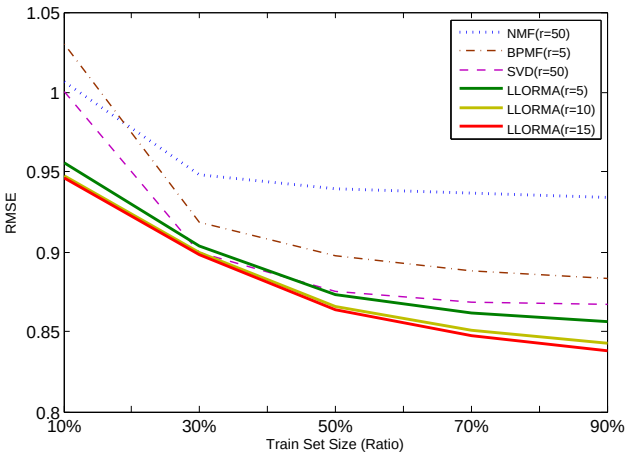

Figure 4. RMSE as a function of train set size for MovieLens 1M data. Local models are indicated by thick solid lines, while other methods are indicated by dotted lines.

anchor points q increases).

Figure 4 graphs the RMSE of LLORMA as a function of the train set size, and compares it with global SVD (rank $r = 50$) and other state-of-the-art baselines: non-negative matrix factorization ($r = 50$) (Lee & Seung, 2001) and Bayesian probabilistic matrix factorization ($r = 5$) (Salakhutdinov & Mnih, 2008b). The test set size was fixed to 10% of the MovieLens 1M and the RMSE was averaged over five train-test splits. The graph shows that all methods improve with the train set size, but LLORMA consistently outperforms SVD and the other baselines.

In summary, we conclude that (1) LLORMA outperforms SVD and other state-of-the-art methods including the Netflix winner and DFC, (2) LLORMA can achieve high performance with a low-rank and only a few anchor points, and (3) LLORMA works well with either a small or a large train set.

8. Related work

Matrix factorization for recommender systems has been researched intensively, especially in the context of the Netflix Prize competition. Billsus & Pazzani (1998) initially proposed applying SVD to CF context. Salakhutdinov & Mnih (2008a) and Salakhutdinov & Mnih (2008b) extended matrix factorization to probabilistic and Bayesian approach, and Lawrence & Urtasun (2009) proposed non-linear version of PMF. Rennie & Srebro (2005) proposed a maximum-margin method. Lee et al. (2012b) conducted a comprehensive experimental study comparing a number of state-of-the-art and traditional recommendation system methods using the PREA toolkit (Lee et al., 2012c).

Recent algorithmic progress in matrix completion was achieved by Toh & Yun (2010); Keshavan et al. (2010). Divide-and-Conquer Matrix Factorization (DFC) (Mackey et al., 2011) also solves a number of smaller matrix factorization problems. Our approach generalizes DFC in that we use a metric structure on $[n_1] \times [n_2]$ and use overlapping partitions. Mirbakhsh & Ling (2013) successfully adopted clustering paradigm for seeking user and item communities.

In addition to single matrix factorization, several ensemble models have been proposed for CF. DeCoste (2006) suggested ensembles of MMMF. The Netflix Prize winner (Bell et al., 2007; Koren, 2008) used combination of memory-based and matrix factorization methods. The Netflix Prize runner-up (Sill et al., 2009) proposed Feature-Weighted Least Square (FWLS), using linear ensemble of learners with dynamic weights. Lee et al. (2012a) extended FWLS by introducing automatic stage-wise feature induction. Kumar et al. (2009); Mackey et al. (2011) applied ensembles to Nystrom method and DFC, respectively.

Related models from the dimensionality reduction literature are Local PCA e.g., (Kambhatla & Leen, 1997) and LLE (Roweis & Saul, 2000). A recent related paper on matrix completion (Wang et al., 2013) applies low-rank factorizations to clusters of points.

Candès & Tao (2010) derived a bound on the performance of low-rank matrix completion, and Candès & Plan (2010) adapted the analaysis to a noisy setting. Related results are obtained by (Shalev-Shwartz et al., 2011; Foygel & Srebro, 2011; Foygel et al., 2012).

9. Summary

We presented a new low-rank matrix approximation based on the assumption that the matrix is locally low-rank. Our proposed algorithm, called LLORMA, is highly parallelizable and thus scales well with the amount of observations and the dimension of the problem. Our experiments indicate that LLORMA outperforms several recent state-of-the-art methods without a significant computational overhead. Our formal analysis generalize standard compressed sensing results. We analyze the performance of LLORMA in terms of its dependency on the matrix size, training set size, and locality (kernel bandwidth parameter). Our method is applicable beyond recommendation systems so long as the locality assumption holds. We thus plan to investigate applications in other domains such as signal and image denoising.

Acknowledgement

We would like to thanks Samy Bengio and Le Song for insightful comments. The work of Guy Lebanon was conducted as a visiting scientist at Google.

References

Bell, R., Koren, Y., and Volinsky, C. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proc. of the ACM SIGKDD Conference*, 2007.

Billsus, D. and Pazzani, M. J. Learning collaborative information filters. In *Proc. of the International Conference on Machine Learning*, 1998.

Candès, E.J. and Plan, Y. Matrix completion with noise. *Proc. of the IEEE*, 98(6):925–936, 2010.

Candès, E.J. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

DeCoste, D. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *Proc. of the International Conference on Machine Learning*, 2006.

Foygel, R. and Srebro, N. Concentration-based guarantees for low-rank matrix reconstruction. *ArXiv Report arXiv:1102.3923*, 2011.

Foygel, R., Srebro, N., and Salakhutdinov, R. Matrix reconstruction with the local max norm. *ArXiv Report arXiv:1210.5196*, 2012.

Kambhatla, N. and Leen, T. K. Dimension reduction by local principal component analysis. *Neural Computation*, 9:1493–1516, 1997.

Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 99:2057–2078, 2010.

Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. of the ACM SIGKDD Conference*, 2008.

Kumar, S., Mohri, M., and Talwalkar, A. Ensemble nystrom method. In *Advances in Neural Information Processing Systems*, 2009.

Lawrence, N. D. and Urtasun, R. Non-linear matrix factorization with gaussian processes. In *Proc. of the International Conference on Machine Learning*, 2009.

Lee, D. and Seung, H. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, 2001.

Lee, J., Sun, M., Kim, S., and Lebanon, G. Automatic feature induction for stagewise collaborative filtering. In *Advances in Neural Information Processing Systems*, 2012a.

Lee, J., Sun, M., and Lebanon, G. A comparative study of collaborative filtering algorithms. *ArXiv Report 1205.3193*, 2012b.

Lee, J., Sun, M., and Lebanon, G. PREA: Personalized recommendation algorithms toolkit. *Journal of Machine Learning Research*, 13:2699–2703, 2012c.

Mackey, L. W., Talwalkar, A. S., and Jordan, M. I. Divide-and-conquer matrix factorization. In *Advances in Neural Information Processing Systems*, 2011.

Mirbakhsh, N. and Ling, C. X. Clustering-based matrix factorization. *ArXiv Report arXiv:1301.6659*, 2013.

Rennie, J.D.M. and Srebro, N. Fast maximum margin matrix factorization for collaborative prediction. In *Proc. of the International Conference on Machine Learning*, 2005.

Roweis, S. and Saul, L. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290: 2323–2326, 2000.

Salakhutdinov, R. and Mnih, A. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, 2008a.

Salakhutdinov, R. and Mnih, A. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proc. of the International Conference on Machine Learning*, 2008b.

Shalev-Shwartz, S., Gonen, A., and Shamir, O. Large-scale convex minimization with a low-rank constraint. In *Proc. of the International Conference on Machine Learning*, 2011.

Sill, J., Takacs, G., Mackey, L., and Lin, D. Feature-weighted linear stacking. *Arxiv preprint arXiv:0911.0460*, 2009.

Toh, K.C. and Yun, S. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6(15):615–640, 2010.

Wand, M. P. and Jones, M. C. *Kernel Smoothing*. Chapman and Hall/CRC, 1995.

Wang, Y., Szlam, A., and Lerman, G. Robust locally linear analysis with applications to image denoising and blind inpainting. *SIAM Journal on Imaging Sciences*, 6(1):526–562, 2013.