Simple and practical algorithms for ℓ_p-norm low-rank approximation

Anastasios Kyrillidis, Dept. of Computer Science, Rice University anastasios@rice.edu

Low-rank optimization - Background

Consider the following optimization problem:

$$\text{(LR)} \qquad \min_{X\in\mathbb{R}^{m\times n}} \ f(X) \quad \text{subject to} \quad \operatorname{rank}(X)\leq r.$$

Common assumptions:

- $X^\star$: (approx.) low rank, *i.e.,* $\operatorname{rank}(X^\star)=r$ *or* $\|X^\star - X_r^\star\|_F$ *small.*
- $f:\mathbb{R}^{m\times n}\to\mathbb{R}$ has (restricted) Lipschitz continuous gradients:

$$\|\nabla f(X)-\nabla f(Y)\|_F \leq L\cdot\|X-Y\|_F, \quad X,Y\in\mathbb{R}^{m\times n}$$

- f might also be (restricted) strongly convex: *f is usually differentiable*

$$f(Y)\geq f(X)+\langle\nabla f(X), Y-X\rangle+\tfrac{\mu}{2}\|Y-X\|_F^2, \quad X,Y\in\mathbb{R}^{m\times n}$$

Why low rankness?

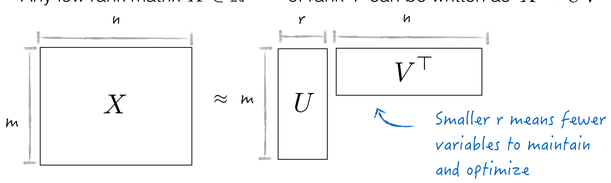
In modern machine learning, one is interested in finding/learning a **low rank** matrix (Occam's razor):

- Low-rankness leads to better low-dimensional recovery.
- Low-rankness results into better computational complexity (fewer variables to optimize).
- Low-rankness might prevent over-fitting in machine learning problems.

Examples: matrix completion, phase retrieval, QST, etc.

Low rankness in factored form to reduce # of variables

Any low rank matrix $X\in\mathbb{R}^{m\times n}$ of rank r can be written as $X=UV^\top$



(LR) reduces to:

$$\text{(Bilinear LR)} \qquad \min_{U\in\mathbb{R}^{m\times r},V\in\mathbb{R}^{n\times r}} \ f(UV^\top) \quad \text{where} \quad r\leq\{m,n\}.$$

Current state of the art

Non-convex!!!

- ***Bi-Factored Gradient Descent*** (BFGD) algorithm:

$$U_{i+1}=U_i-\eta\nabla f(U_iV_i^\top)\cdot U_i$$
$$V_{i+1}=V_i-\eta\nabla f(U_iV_i^\top)^\top\cdot V_i$$

- (Local) convergence of BFGD under common assumptions:
  - **Linear convergence** for smooth and strongly convex f.
  - **Sublinear** convergence for just smooth f.

$$f(U_TV_T^\top)-f(X^\star)\leq\frac{10\|U_0V_0^\top-X^\star\|_F^2}{\eta T}$$

What is this paper about?

We consider the following factorization problem:

$$(\ell_p\text{-LR}) \qquad \min_{U,V}\|M-UV^\top\|_p, \quad p\in\{1,\infty\}$$

(Norms are element-wise norms)

Applications with ℓ_1- and $\ell_\infty-$norms:

- ℓ_1-norm: *robust PCA, background subtraction, motion detection, detection of brain activation patterns, detection of anomalous behavior in dynamic networks, etc.*
- $\ell_\infty-$ norm: *Maximal volume concept in matrix approximation, skeleton approximations, low-rank recovery from quantized measurements.*

Issues with ℓ_p-LR

- **No straightforward closed-form solution:** while Frobenius norm LR has a closed form solution (via SVD), this does not hold for $p\in\{1,\infty\}$.
- **No hope for exact solution in polynomial time:** In fact, $\ell_1-$and $\ell_\infty-$ norm low-rank approximations are NP-hard.
- **Thus only approximations is the best we can hope for.**

Our approach and contributions

Smoothing technique

+

Recent results on $UV^\top$ matrix factorization

=

$$\|M-\widehat{U}\widehat{V}^\top\|_p\leq(1+\varepsilon)\cdot\mathrm{OPT}$$

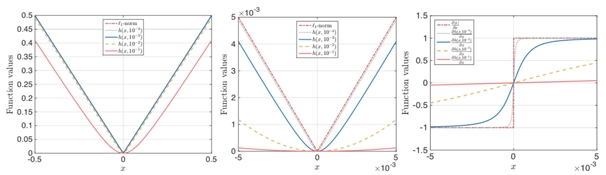
- Polynomial approximation algorithm that achieves $(1+\varepsilon)$-approximation guarantee.
- Theory requires *a priori* knowledge of ground-truth-related quantities to achieve this *(Achilles heel but also indicative of what is needed to be known for good approximation).*
- The proposed scheme has a simple and practical implementation, that outpeforms state of the art.

Charbonnier approximation and `logsumexp`

To use BFGD, f should be at least once differentiable.

· Approximating the entry wise ℓ_1-norm:

$$h(X,\tau)=\tau\cdot\sum_{i=1}^m\sum_{j=1}^n\left(\sqrt{\left(\frac{X_{ij}}{\tau}\right)^2+1}-1\right) \quad \textbf{(Charbonnier loss function)}$$



Some key properties:

Lemma 4.1. *For any $X\in\mathbb{R}^{m\times n}$:*

- $\nabla h(X,\tau)=\frac{1}{\tau}X\odot S\in\mathbb{R}^{m\times n}$, *where* $S\in\mathbb{R}^{m\times n}$ *and* $S_{ij}:=\frac{2}{\sqrt{(X_{ij}/\tau)^2+1}}$,
- $\nabla^2 h(X,\tau)=\frac{1}{\tau}I\odot Q\in\mathbb{R}^{mn\times mn}$, *where* $Q\in\mathbb{R}^{mn\times mn}$ *and* $Q_{ij}:=\frac{2}{\left((X_{ij}/\tau)^2+1\right)^{3/2}}$.

Lemma 4.2. *Function h is a convex continuously differentiable function and it has Lipschitz continuous gradients with constant $\frac{2}{\tau}$. Moreover:*

$$|X|_1-mn\tau\leq h(X,\tau)\leq|X|_1.$$

· Approximating the entry wise $\ell_\infty-$norm:

$$\sigma(X,\tau)=\tau\cdot\log\left(\frac{\sum_{i=1}^m\sum_{j=1}^n e^{X_{ij}/\tau}+e^{-X_{ij}/\tau}}{2mn}\right) \quad \textbf{(Logsumexp function)}$$

Some key properties:

Lemma 4.3. *For any $X\in\mathbb{R}^{m\times n}$:*

- $\nabla\sigma(X,\tau)=\frac{1}{\mathrm{Tr}(\mathbb{1}\cdot P)}\cdot N\in\mathbb{R}^{m\times n}$,
- $\nabla^2\sigma(X,\tau)=\frac{\left(diag(vec(P))-\frac{vec(N)vec(N)^\top}{\mathrm{Tr}(\mathbb{1}\cdot P)}\right)}{\tau\cdot\mathrm{Tr}(\mathbb{1}\cdot P)}\in\mathbb{R}^{mn\times mn}$

where $\mathtt{diag}(\cdot):\mathbb{R}^{mn}\to\mathbb{R}^{mn\times mn}$ turns the vector input to a diagonal matrix output, $\mathtt{vec}(\cdot):\mathbb{R}^{m\times n}\to\mathbb{R}^{mn}$ turns a matrix to a vector by "stacking" its columns, and $\mathbb{1}$ denotes the all-ones matrix.

Lemma 4.4. *The* `logsumexp` *function σ is a convex continuously differentiable function and it has Lipschitz continuous gradients with constant $\frac{1}{\tau}$. Moreover:*

$$|X|_\infty-\tau\log(2mn)\leq\sigma(X,\tau)\leq|X|_\infty.$$

Approximate solver for ℓ_p-LR

We propose the following surrogate optimization problems:

$$\min_{U,V}h(M-UV^\top,\tau)+\frac{\lambda}{2}\|UV^\top\|_2^2 \qquad (\ell_1-\text{norm})$$
$$\min_{U,V}\sigma(M-UV^\top,\tau)+\frac{\lambda}{2}\|UV^\top\|_2^2 \qquad (\ell_\infty-\text{norm})$$

- λ parameter: makes problem *strongly convex,* used for theoretical purposes.

Proposed algorithm

Algorithm 2 ℓ_1-norm low rank approximation solver

Parameters: r, OPT, values of $|X^\star|_2^2$ and $\sigma_r(\widehat{X}^\star)$, $\varepsilon>0$.
Set $\tau=\frac{\varepsilon\cdot\mathrm{OPT}}{3mn}$.
Set function $T=O\left(\frac{\sigma_r(\widehat{X}_r^\star)}{\varepsilon\mathrm{OPT}}\right)$.
Set $\lambda=\frac{2\varepsilon\cdot\mathrm{OPT}}{3|X^\star|_2^2}$
Compute $\widehat{L}=(\frac{1}{\tau}+\lambda)$.
Set $f(UV^\top):=h(M-UV^\top,\tau)+\frac{\lambda}{2}|UV^\top|_2^2$.
Run Algorithm 1 $(U_T,\ V_T)=\mathtt{BFGD}(r,T,\frac{1}{4},1,\widehat{L})$.

(Similar algorithm for $\ell_\infty-$norm low-rank approximation.)

Guarantees

· For ℓ_1-norm:

$$|M-U_TV_T^\top|_1\leq(1+\varepsilon)\cdot\mathrm{OPT},$$
$$\textit{after } T=O\left(\sigma_r(\widehat{X}_r^\star)\left(\frac{mn}{(\varepsilon\mathrm{OPT})^2}+\frac{1}{\|X^\star\|_2^2}\right)\right)\textit{ iterations.}$$

· For $\ell_\infty-$norm:

$$|M-U_TV_T^\top|_\infty\leq(1+\varepsilon)\cdot\mathrm{OPT},$$
$$\textit{after } T=O\left(\sigma_r(\widehat{X}_r^\star)\left(\frac{\log(mn)}{(\varepsilon\mathrm{OPT})^2}+\frac{1}{\|X^\star\|_2^2}\right)\right)\textit{ iterations.}$$

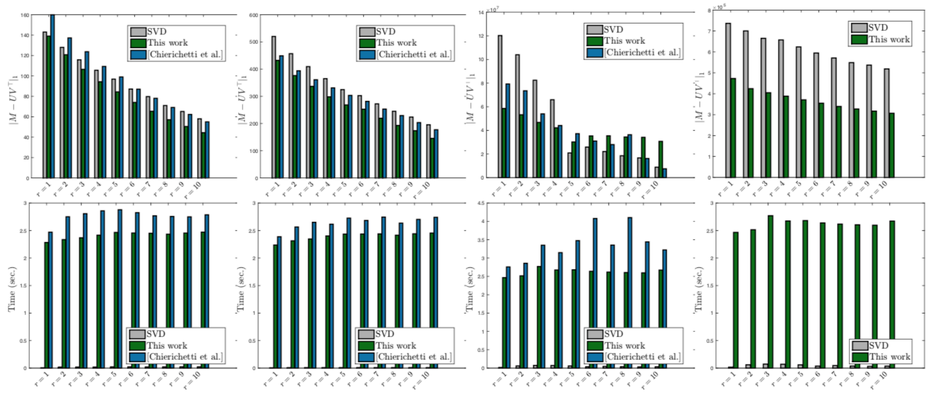


Figure 2: Top row: function value performance $|M-UV^\top|_1$; Bottom row: corresponding execution time. In all settings, we set problem (1) for $r=\{1,\dots,10\}$. First column: $M\in[0,1]^{20\times30}$ where each entry is randomly and independently generated. Second column: $M\in\{-1,1\}^{20\times30}$ where each entry is randomly and independently generated. Third column: $M\in\mathbb{R}^{27\times27}$ is the FIDAP matrix. Fourth column: $M\in\mathbb{R}^{3430\times6906}$ is the word-frequency matrix. In the latter case, the sub-solver for ℓ_p-projection was not able to complete the task, and thus the algorithm in [9] is omitted.