Approximate Low-Rank Tensor Learning

Yaoliang Yu
Dept of Machine Learning
Carnegie Mellon University
yaoliang@cs.cmu.edu

Hao Cheng
Dept of Electric Engineering
University of Washington
kelvinwsch@gmail.com

Xinhua Zhang
Machine Learning Group
NICTA and ANU
xizhang@nicta.com.au

Abstract

Many real-world data arise naturally in the form of tensors, i.e. multi-dimensional arrays. Equipped with an appropriate notion of low-rankness, learning algorithms can benefit greatly from exploiting the rich dependency encoded in a tensor. However, due to complexity issues, most existing works still resort to unfolding the tensor to one (or several) matrix, at the risk of losing valuable structural information. To address this problem, we choose to directly learn a low-rank tensor in an *approximate* manner. By combining a simple approximation algorithm for the tensor spectral norm with the recent generalized conditional gradient, we establish a formal optimization guarantee for a general low-rank tensor learning formulation. Extensive experiments verify the superiority of our algorithm.

1 Introduction

Real-world data usually exhibit rich structures that learning algorithms can significantly benefit from. While data in matrix forms may have low rank, more general multi-way correlations are often observed in tensor structured data, i.e. multi-dimensional array [1]. Example applications include multi-channel images, video sequences, chemical compound processes, antenna array signals, etc.

Remarkable success has been achieved in low-rank matrices since the seminal work of [2]. Naturally, one expects a similar picture for tensors—after all they are "just" high order matrices. Unfortunately, the genuine extension of the matrix rank to tensors is challenging, because many "obvious" properties in matrices cease to hold true. Further complications arise from complexity issues, e.g. computing the tensor rank is NP-Hard [3]. As a result, many existing methods first unfold the tensor into a matrix along a certain mode, and then apply the existing matrix technique, e.g., minimizing the matrix trace norm as a proxy of the rank. In order to complete a partially observed tensor, [4] compared three methodologies: complete slice-by-slice, take the best of the mode-k matrix completions, and use a weighted sum of mode-k trace norms. [4] found that the last strategy worked best in a variety of tasks. Similar ideas also appeared in [5–8]. Instead of enforcing low rank in all modes, [9] considered decomposing a tensor into a mixture of tensors, each employing low rank in a specific mode. Recently, [10] proposed a relaxation that is tighter than the matrix trace norm.

However, unfolding a genuine tensor into a matrix may be unsatisfying or even misleading. Intuitively, a tensor of "low" rank can nevertheless have large or even full rank in all matrix unfoldings. Therefore in this paper, we propose taking a different approach by learning a low-rank tensor through *directly* minimizing the tensor trace norm, an effective convex surrogate suggested in the atomic norm framework [11]. Recently this principle has also been shown promising in theory [12].

Unfortunately tensor trace norm is intractable in computation [3]. To overcome the complexity barrier, we forgo exact solutions and resort to approximate solutions which provide sub-optimality guarantees. This idea is consistent with the standard practice of addressing NP-Hard problems with approximate algorithms that are provably "good". However, we are not aware of such algorithms in the context of tensor trace norm optimization[1], or efficient implementations that scale up to large data. Therefore, our key contribution is to fill this gap by utilizing a simple approximation algorithm

[1]For example, [12] did not provide any computational algorithm.

for computing the tensor *spectral* norm, within the optimization framework of generalized conditional gradient [GCG, 13, 14]. By developing a novel error bound for GCG based on *multiplicatively* approximate oracle, we further prove the approximation guarantee of our optimization method for a general formulation of low-rank tensor learning. Thanks to the low-rank factorization of the solution that our approach maintains explicitly, the per-step complexity and memory consumption are kept low, allowing it to scale to large datasets. Extensive experiments consistently corroborate the superior efficiency and accuracy of our scheme, in comparison to state-of-the-art competitors.

It is noteworthy that a large body of works exist which directly find a low-rank tensor decomposition, e.g. [1, 15–17]. However, they usually suffer from a) commitment to a pre-selected rank parameter; b) tailored for the specific least squares loss; and c) weak convergence guarantee. In contrast, we *promotes* low-rank for any loss through the tensor trace norm, finding an optimal rank automatically.

2 CP Decomposition and Low-Rank Tensor Learning

We identify a tensor as a multi-dimensional array $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$, where K is the order and each I_k is the dimension of the k-th mode. Following [1], we define the mode-k multiplication with respect to any matrix $U \in \mathbb{R}^{J_k \times I_k}$ as $\mathcal{A} \times_k U \in \mathbb{R}^{I_1 \times \ldots \times I_{k-1} \times J_k \times I_{k+1} \times \ldots \times I_K}$ with elements

$$(\mathcal{A} \times_k U)_{i_1,\ldots,i_{k-1},j_k,i_{k+1},\ldots,i_K} = \sum\nolimits_{i_k=1}^{I_k} \mathcal{A}_{i_1,\ldots,i_{k-1},i_k,i_{k+1},\ldots,i_K} U_{j_k,i_k}. \tag{1}$$

As usual, we define the inner product between two tensors with the same size as $\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{i_1,\ldots,i_K} \mathcal{A}_{i_1,\ldots,i_K} \mathcal{B}_{i_1,\ldots,i_K}$, and the induced Frobenius norm $\|\mathcal{A}\|_{\mathsf{F}} := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. We can unfold (or flatten) a tensor into a 2-D matrix as follows. For any $1 \le k \le K$, $\mathcal{A}_{(k)} \in \mathbb{R}^{I_k \times (\prod_{j \ne k} I_j)}$, with its $\big(i_k, 1 + \sum_{j=1, j\ne k}^{K} (i_j - 1) \prod_{m=j+1, m\ne k}^{K} I_m\big)$-th entry being $\mathcal{A}_{i_1,\ldots,i_K}$. Like matrices, we can decompose a tensor into a combination of *primitives*, in particular, low-rank factors. We call the tensor $\mathcal{A}$ rank-1 if it can be written as the tensor outer product $\mathcal{A} = \mathbf{u} \circ \mathbf{v} \circ \cdots \circ \mathbf{z}$, where the K vectors $\mathbf{u} \in \mathbb{R}^{I_1}, \ldots, \mathbf{z} \in \mathbb{R}^{I_K}$. That is, $\mathcal{A}_{i_1,\cdots,i_K} = u_{i_1} v_{i_2} \cdots z_{i_K}$. We say $\operatorname{rank}(\mathcal{A}) = r$ if $\mathcal{A}$ can be decomposed into a sum of r (but not less) rank-1 tensors, namely $\mathcal{A} = \sum_{i=1}^{r} \mathbf{u}_i \circ \mathbf{v}_i \circ \cdots \circ \mathbf{z}_i$. This decomposition is known as CANDECOMP/PARAFAC, or more succinctly CP.

In general $\operatorname{rank}(\mathcal{A})$ can be *much* larger than $\max_k \operatorname{rank}(\mathcal{A}_{(k)})$ or $\max_k I_k$. Even when $\operatorname{rank}(\mathcal{A})$ is low (compared to its *degree of freedom*), the unfolding $\mathcal{A}_{(k)}$ can have large or even full matrix rank for all k, which suggests that treating a tensor as unfolded matrices can be unsatisfying. Computing the CP decomposition, or the rank, is intractable [3]. A popular approximation is the alternating least squares (ALS), which alternatively optimizes each factor matrix with all others fixed [1].

Mathematically, we aim at solving the following optimization problem:

$$\min_{\mathcal{W}} \ \ell(\mathcal{W}) + \lambda \cdot \operatorname{rank}(\mathcal{W}), \tag{2}$$

where ℓ is any proper convex loss with Lipschitz continuous gradient. In tensor completion, $\ell(\mathcal{W}) = \|\mathscr{L}(\mathcal{W} - \mathcal{Z})\|_{\mathsf{F}}^2$, where $\mathcal{Z}$ is some observed tensor, and $\mathscr{L} : \mathbb{R}^{I_1 \times \cdots \times I_K} \to \mathbb{R}^{I_1 \times \cdots \times I_K}$ is some linear operator (e.g. the sampling operator). Here we are interested in the genuine tensor rank, rather than the mode-k rank. As mentioned in the introduction, we believe and verify, through extensive experiments, that pursuing a small tensor rank is meaningful and potentially advantageous in practice. To address the computational issues of tensor rank, we follow the principle of convex relaxation for matrix ranks, and pursue tensor trace norms that [11] justified in a general atomic norm framework. Terefore we consider the regularized risk minimization

$$\min_{\mathcal{W}} \ell(\mathcal{W}) + \lambda \cdot \|\mathcal{W}\|_{\mathrm{tr}}, \tag{3}$$

where $\|\cdot\|_{\mathrm{tr}}$ is the tensor trace norm [TNN, 11, 18, 19], defined as the dual of the spectral norm:

$$\|\mathcal{A}\|_{\mathrm{sp}} := \max\{\langle \mathcal{A}, \mathbf{u}_1 \circ \cdots \circ \mathbf{u}_K \rangle : \ \|\mathbf{u}_i\|_2 \le 1, \ i = 1, \ldots, K\}. \tag{4}$$

Obviously, $\|\mathcal{A}\|_{\mathrm{sp}} \le \big\|\mathcal{A}_{(k)}\big\|_{\mathrm{sp}}$, implying that $\|\mathcal{A}\|_{\mathrm{tr}} \ge \big\|\mathcal{A}_{(k)}\big\|_{\mathrm{tr}}$. For $K \ge 3$, almost no significant effort has been made on solving (3), because the tensor trace norm itself is intractable to compute [3]. However, (3) does enjoy some advantages besides convexity. [12] proved that the tensor trace norm leads to improved sample size requirements. Furthermore, [11] managed to reformulate (3) as a hierarchy of semidefinite programs (SDP). In theory, as the hierarchy approaches infinite, one could solve (3) optimally. But SDP is expensive and hard to scale to large datasets. Our main contribution, presented in the next section, is a simple algorithm that *approximately* solves (3) with sub-optimality guarantee, and scales efficiently on large datasets.

Algorithm 1: Approximate generalized conditional gradient for sublinear κ

choose $\mathbf{w}_1 \in \operatorname{dom}\ell \cap \operatorname{dom}\kappa$.
for $t = 1, 2, \ldots$ **do**
  Compute $\mathbf{g}_t \leftarrow \nabla\ell(\mathbf{w}_t)$, and find $\mathbf{z}_t \in \mathsf{B}_\kappa$ so that $\langle \mathbf{z}_t, \mathbf{g}_t\rangle \le \alpha \cdot \min_{\mathbf{z}\in\mathsf{B}_\kappa}\langle \mathbf{z}, \mathbf{g}_t\rangle$;
  $\eta_t \leftarrow 2/(t+2)$; $s_t \leftarrow \operatorname{argmin}_{s\ge 0}\ell((1-\eta_t)\mathbf{w}_t + s\eta_t\mathbf{z}_t) + \lambda\cdot s\eta_t$;
  $\tilde{\mathbf{w}}_{t+1} \leftarrow (1-\eta_t)\mathbf{w}_t + \eta_t(s_t\mathbf{z}_t)$;
  choose $\mathbf{w}_{t+1}$ so that $f(\mathbf{w}_{t+1}) \le \min\{f(\mathbf{w}_t), f(\tilde{\mathbf{w}}_{t+1})\}$.

3 Approximate Generalized Conditional Gradient

Our algorithm is based on the recent work of generalized conditional gradient [GCG, 13, 14], which extends the conditional gradient (a.k.a. Frank-Wolfe [20]) to the penalized objective in the form of (3). Our results below allow the trace norm to be generalized into any positive homogeneous convex function κ, and the tensor variable into any Hilbert space. So henceforce we simply write κ in place of $\|\cdot\|_{\text{tr}}$, and boldface symbol $\mathbf{w}$ in place of curly $\mathcal{W}$. The basic algorithm of [13, 14] successively linearizes the loss ℓ at the current iterate $\mathbf{w}_t$, finds an update direction from the "unit ball" B_κ:

$$\mathbf{z}_t \in \operatorname{argmin}_{\mathbf{z}\in\mathsf{B}_\kappa}\langle \mathbf{z}, \nabla\ell(\mathbf{w}_t)\rangle, \quad \text{where} \quad \mathsf{B}_\kappa := \{\mathbf{w} : \kappa(\mathbf{w}) \le 1\}, \tag{5}$$

and then performs the update $\mathbf{w}_{t+1} = (1-\eta_t)\mathbf{w}_t + \eta_t s_t\mathbf{z}_t$ with some step size η_t and scaling factor s_t. However, when GCG is applied to (3), the problem (5) requires computing the tensor spectral norm (4), which is known to be intractable for $K \ge 3$ [3] . The crucial observations we make here are a) like many other NP-Hard problems, the tensor spectral norm admits simple approximation algorithms; and b) GCG is "robust" against approximate subroutines—a point we demonstrate first.

Theorem 1 *Let $\ell \ge 0$ be convex, smooth, and have bounded sublevel sets; κ be a positive homogeneous convex function. Then after t iterations Algorithm 1 outputs $\mathbf{w}_{t+1}$ such that for all $\mathbf{w}$,*

$$f(\mathbf{w}_{t+1}) - \frac{f(\mathbf{w})}{\alpha} \le \frac{4C}{t+2}, \quad \textit{where} \quad f(\mathbf{w}) = \ell(\mathbf{w}) + \lambda\kappa(\mathbf{w}).$$

Here C is some absolute constant depending on f, $\mathbf{w}$, $\mathbf{w}_1$ and α. (Proof is provided in [21]).

The approximation constant α lies in $]0, 1]$, because the objective in the subroutine (5) is nonpositive (the ball B_κ contains the origin). Note that [22] considered a different multiplicative-approximate scheme, which in essence requires, in each step, an approximate solution of $\min_{\mathbf{z}\in\mathsf{B}_\kappa}\langle \mathbf{z}, \nabla\ell(\mathbf{w}_t)\rangle - \langle \mathbf{w}_t, \nabla\ell(\mathbf{w}_t)\rangle$. Unfortunately, due to the last constant term, it is usually hard, if possible, to come up with a multiplicative approximate solution for it. On the other hand, [22] was still able to prove *exact* convergence while Theorem 1 here yields only an approximate guarantee. We noted in passing that a strategy similar to ours appeared independently in [23] for matrix factorization.

Approximately Computing the Tensor Spectral Norm We still need to solve the subroutine (5), approximately. Fortunately tensor spectral norm admits a particularly simple approximation algorithm [24]. The idea is to iteratively reduce the size of the tensor whereas in the k-th step we lose at most a factor of $1/\sqrt{I_k}$ (due to relaxing the Frobenius norm to the spectral norm). Algorithm 2 summarizes the main steps. Note that we only need to compute the spectral norm (as opposed to trace norm) of the unfolded matrices $\mathcal{A}_{(k)}$, which can be advantageous in large applications.

It is straightforward to prove that Algorithm 2 yields at least an $\alpha = \prod_{k=1}^{K-2} I_k^{-1/2}$-approximate solution where $I_1 \le \cdots \le I_K$ (see proof in [21]). Combining Algorithm 1 and 2 we immediately obtain an $\alpha = \prod_{k=1}^{K-2} I_k^{-1/2}$-approximate solution for the low-rank tensor learning problem (3). Although the approximation ratio depends on the size (skipping the two largest modes), it is only a worst-case bound which is likely not tight. Indeed, by enhancing Algorithm 2 with some local search procedure such as ALS, in experiments we almost always find the optimal solution. We mention that it is possible to get a (slightly) better approximation bound, at the expense of complicating the algorithm [25]. Here, we adopt Algorithm 2 mostly because of its simplicity and efficiency.

Practical Acceleration A very effective acceleration strategy was proposed in [13] for the matrix trace norm, and it can be extended to the tensor setting by utilizing a new variational representation of the trace norm (see [21] for the proof, and contrasts with other tensor regularizers in [7, 26]):

$$\|\mathcal{A}\|_{\text{tr}} = \inf\left\{\tfrac{1}{K}\textstyle\sum_i \|\mathbf{u}_i\|_2^K + \|\mathbf{v}_i\|_2^K + \cdots \|\mathbf{z}_i\|_2^K \;:\; \mathcal{A} = \sum_i \mathbf{u}_i \circ \mathbf{v}_i \circ \cdots \circ \mathbf{z}_i\right\}. \tag{6}$$

Algorithm 2: Approximate computation of tensor spectral norm for $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots I_K}$. $I_1 \leq \cdots \leq I_K$.

for $k = K-1, \ldots, 2$ **do**
  $\mathbf{u}_k \leftarrow$ top left singular vector of $\mathcal{A}_{(k)}$, and flatten the k-th mode of $\mathcal{A}$ by $\mathcal{A} \leftarrow \mathcal{A} \times_k \mathbf{u}_k^\top$;
$(\mathbf{u}_1, \mathbf{u}_K) \leftarrow$ top left and right singular vectors of $\mathcal{A}_{(1)}$.

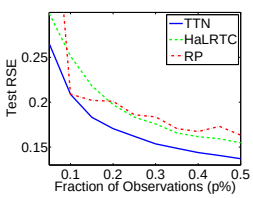

Figure 1: RSE vs p for image inpainting

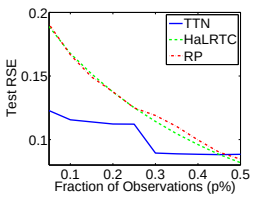

Figure 2: RSE vs p for video completion

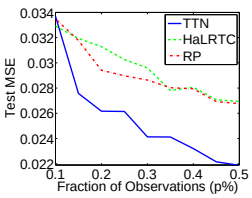

Figure 3: RSE vs p for school grade prediction

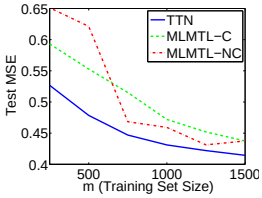

Figure 4: Test MSE vs p for restaurant rating

After the t-th iteration, we obtain a low-rank representation of the current iterate $\mathcal{W}_t = \sum_{i=1}^t \mathbf{u}_i \circ \cdots \circ \mathbf{z}_i$. Using (6), we may interlace GCG updates with optimization of a surrogate objective:

$$\min_{\tilde{\mathbf{u}}_i, \ldots, \tilde{\mathbf{z}}_i} \ \ell(\Sigma_{i=1}^t \tilde{\mathbf{u}}_i \circ \cdots \circ \tilde{\mathbf{z}}_i) + \frac{\lambda}{K} \sum_{i=1}^t \left(\|\tilde{\mathbf{u}}_i\|_2^K + \cdots \|\tilde{\mathbf{z}}_i\|_2^K \right), \tag{7}$$

initialized with the t-th iterate of Algorithm 1. The resulting procedure is in [21]. Importantly, (7) is easy to evaluate, facilitating local optimization via, e.g., L-BFGS. Although a naive implementation of gradient computation costs $O(K\Pi_k I_k)$ time, it can be reduced by a factor of K via dynamic programming [21]. Empirically, the seamless integration of (7) significantly speeds up the convergence of GCG, and this is enabled by the explicit low-rank representation maintained in Algorithm 1.

4 Experiments

We studied the empirical performance of TTN in tensor completion, robust PCA, and multitask learning. The complete results are given at [21], and we highlight some of them due to space limits.

Low-rank Tensor Completion We first consider the loss $\|\mathscr{L}(\mathcal{W} - \mathcal{Z})\|_{\mathsf{F}}^2$ where $\mathscr{L}$ is the mask operator from $\mathbb{R}^{I_1 \times \cdots \times I_K}$ to $\mathbb{R}^{I_1 \times \cdots \times I_K}$, which simply fills the unobserved entries with zero. Two state-of-the-art algorithms were used for comparison: HaLRTC [4] which uses the weighted trace norm on unfolded matrices; and a tighter relaxation by Romera-Paredes and Pontil [10], referred as RP. For all methods, we randomly picked $p\%$ elements of $\mathcal{Z}$ as observations for training (20 random repetitions). The value of λ was selected via a validation set, which always consisted of 10% of the whole dataset. The test error was evaluated on the remaining $(90-p)\%$ entries.

Figures 1 to 3 show the test root square error (RSE) as a function of p, over three real datasets. The image inpainting dataset uses `facade` from [4], representing images as a "width (259)" $\times$ "height (247)" $\times$ "RGB (3)" tensor. Both RSE and visual results demonstrate the superiority of TTN in propagating global structure from a small number of observed pixels. The video completion task used the Ocean video [4, 10], where videos form 4-order tensors sized $160 \times 112 \times 3 \times 32$ ("width" $\times$ "height" $\times$ "RGB" $\times$ "frame"). Figure 2 shows that TNN is again more effective than other competitors. Following [10], we used the Inner London Education Authority dataset, representing each of the 15,362 students with five categorical attributes: school, gender, VR-band, ethnic, and year. This led to a 5-order tensor $\mathcal{Z} \in \mathbb{R}^{139 \times 2 \times 3 \times 11 \times 3}$. The prediction of exam scores was therefore formulated as a tensor completion problem. Figure 3 confirms that lower error is achieved by TTN.

Multitask Learning Our last experiment uses a low-rank tensor for multitask learning in restaurant recommendation [7]. It predicts the rating that a restaurant (represented by an $I_1 = 45$ dimensional feature vector) would receive from each of the $I_2 = 138$ customers, in $I_3 = 3$ aspects (food, service, overall). So there are $I_2 \times I_3$ tasks, with each task employing a I_1 dimensional weight vector, and the whole weight tensor is assumed to have low rank. We used ℓ_2 loss for ℓ.

Following [7], we randomly sampled m ratings (across all tasks) for training, 10% for validation, and the rest for testing. TTN was compared with the convex multilinear multitask learning model (MLMTL-C) and its non-convex variant MLMTL-NC, which, as shown by [7], predicts more accurately than a number of other multitask algorithms. The test MSE as a function of m is shown in Figure 4. Clearly, TTN yields significantly lower test error over a range of training set sizes.

References

[1] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51:455–500, 2009.

[2] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[3] C. J. Hillar and L.-H. Lim. Most tensor problems are NP-hard. *Journal of ACM*, 60(6):45:1–45:39, 2013.

[4] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

[5] M. Signoretto, L. D. Lathauwer, and J. A. K. Suykens. Nuclear norms for tensors and their use for convex multilinear estimation. *Tech. rep.*, 2010.

[6] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27:1–19, 2011.

[7] B. Romera-Paredes, M. H. Aung, N. Bianchi-Berthouze, and M. Pontil. Multilinear multitask learning. *In ICML*. 2013. URL `https://sites.google.com/site/romeraparedes/code/`.

[8] D. Goldfarb and Z. Qin. Robust low-rank tensor recovery: Models and algorithms. *SIAM Journal on Matrix Analysis and Applications*, 35:225–253, 2014.

[9] R. Tomioka and T. Suzuki. Convex tensor decomposition via structured schatten norm regularization. *In NIPS*. 2013.

[10] B. Romera-Paredes and M. Pontil. A new convex relaxation for tensor completion. *In NIPS*. 2013.

[11] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S.Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[12] M. Yuan and C.-H. Zhang. On tensor completion via nuclear norm minimization, 2014.

[13] X. Zhang, Y. Yu, and D. Schuurmans. Accelerated training for matrix-norm regularization: A boosting approach. *In NIPS*. 2012.

[14] Z. Harchaoui, A. Juditsky, and A. Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 2014.

[15] L. de Lathauwer, B. de Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[16] J. Nie and L. Wang. Semidefinite relaxations for best rank-1 tensor approximations, 2013.

[17] B. Jiang, S. Ma, and S. Zhang. Tensor principal component analysis via convex optimization. *Mathematical Programming*, to appear, 2014.

[18] L.-H. Lim and P. Comon. Blind multilinear identification. *IEEE Transactions on Information Theory*, 60(2):1260–1280, 2014.

[19] H. Derksen. On the nuclear norm and the singular value decomposition of tensors, 2014. ArXiv:1308.3860v2.

[20] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, Mar 1956.

[21] Anonymous. Long version of this paper, 2014. `http://1drv.ms/1naS5i0`.

[22] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate frank-wolfe optimization for structural svms. *In ICML*. 2013.

[23] F. Bach. Convex relaxations of structured matrix factorizations, 2013.

[24] Z. Li, S. He, and S. Zhang. *Approximation Methods for Polynomial Optimization: Models, Algorithms, and Applications*. Springer, 2012.

[25] A. M.-C. So. Deterministic approximation algorithms for sphere constrained homogeneous polynomial optimization problems. *Mathematical Programming*, 129:357–382, 2011.

[26] V. D. Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.