Supplemental material of
Coordinate-descent for learning orthogonal matrices through Givens rotations

Uri Shalit URI.SHALIT@MAIL.HUJI.AC.IL
ICNC-ELSC & Computer Science Department, The Hebrew University of Jerusalem, 91904 Jerusalem Israel
The Gonda Brain Research Center, Bar Ilan University, 52900 Ramat-Gan, Israel

Gal Chechik GAL.CHECHIK@BIU.AC.IL
The Gonda Brain Research Center, Bar Ilan University, 52900 Ramat-Gan, Israel

A. Proofs of theorems of Section 3

Below we use a slightly modified definition of Algorithm 1. The difference lies only in the sampling procedure, and is essentially a technical difference to ensure that each coordinate step indeed improves the objective or lies at an optimum, so that the proofs could be stated more succinctly.

Algorithm 4 Riemannian coordinate minimization on $\mathcal{O}_d$, sampling variant

```
Input: Differentiable objective function $f$, initial matrix
  $U_0 \in \mathcal{O}_d$
  $t = 0$
  while not converged do
    1. Sample coordinate pairs $(i(t), j(t))$ such that $1 \leq i(t) < j(t) \leq d$ uniformly at random without replacement, until the objective function can improve
    2. $U_{t+1} = \underset{\theta}{\operatorname{argmin}}\ f\left(U_t \cdot G(i,j,\theta)\right)$.
    3. $t = t + 1$.
  end while
```

Definition 1. *A point $U_* \in \mathcal{O}_d$ is* asymptotically stable *with respect to Algorithm 4 if it has a neighborhood $\mathcal{V}$ of U_* such that all sequences generated by Algorithm 4 with starting point $U_0 \in \mathcal{V}$ converge to U_*.*

Theorem 1. Convergence to local optimum
(a) The sequence of iterates U_t of Algorithm 4 satisfies: $\lim_{t\to\infty} ||\nabla f(U_t)|| = 0$. This means that the accumulation points of the sequence $\{U_t\}_{t=1}^{\infty}$ are critical points of f.
(b) Assume the critical points of f are isolated. Let U_* be a critical point of f. Then U_* is a local minimum of f if and only if it is asymptotically stable with regard to the sequence generated by Algorithm 4.

Proof. (a) Algorithm 4 is obtained by taking a step in each iteration t in the direction of the tangent vector Z_t, such that for the coordinates $(i(t), j(t))$ we have $(Z_t)_{ij} = -(\nabla f(U_t))_{ij}$, $(Z_t)_{ji} = -(\nabla f(U_t))_{ji}$, and $(Z_t)_{kl} = 0$ for all other coordinates (k, l).

The sequence of tangent vectors $Z_t \in T_{U_t}\mathcal{O}_d$ is easily seen to be gradient related: $\limsup k \rightarrow \infty \langle \nabla f(U_t), Z_t \rangle < 0$ [1]. This follows from Z_t being equal to exactly two coordinates of $\nabla f(U_t)$, with all other coordinates being 0.

Using the optimal step size as we do assures at least as large an increase $f(U_t) - f(U_{t+1})$ as using the Armijo step size rule (Armijo, 1966; Bertsekas, 1999). Using the fact that the manifold $\mathcal{O}_d$ is compact, we obtain by theorem 4.3.1 and corollary 4.3.2 of Absil et al. (2009) that $\lim_{t\to\infty} ||\nabla f(U_t)|| = 0$

(b) Since Algorithm 4 produces a monotonically decreasing sequence $f(U_t)$, and since the manifold $\mathcal{O}_d$ is compact, we are in the conditions of Theorems 4.4.1 and 4.4.2 of Absil et al. (2009). These imply that the only critical points which are local minima are asymptotically stable.

□

We now provide a rate of convergence proof. This proof is a Riemannian version of the proof for the rate of convergence of Euclidean random coordinate descent for nonconvex functions given by Patrascu & Necoara (2013).

Definition 2. *For an iteration t of Algorithm 4, and a set of indices $(i(t), j(t))$, we define the auxiliary single variable function g_t^{ij} :*

$$g_t^{ij}(\theta) = f\left(U_t \cdot G(i,j,\theta)\right), \tag{1}$$

Note that g_t^{ij} are differentiable and periodic with a period of 2π. Since $\mathcal{O}_d$ is compact and f is differentiable there exists a single Lipschitz constant $L(f) > 0$ for all g_t^{ij}.

[1] To obtain a rigorous proof we slightly complicated the sampling procedure in line 1 of Algorithm 1, such that coordinates with 0 gradient are not resampled until a non-zero gradient is sampled.

Theorem 2. Rate of convergence
Let f be a continuous function with L-Lipschitz directional derivatives [2]. Let U_t be the sequence generated by Algorithm 4. For the sequence of Riemannian gradients $\nabla f(U_t) \in T_{U_t}\mathcal{O}_d$ we have:

$$\max_{0\leq t\leq T} E\left[||\nabla f(U_t)||_2^2\right] \leq \frac{L \cdot d^2\left(f(U_0) - f_{min}\right)}{T+1} \quad . \tag{2}$$

Lemma 1. *Let $g : \mathbb{R} \rightarrow \mathbb{R}$ be a periodic differentiable function, with period 2π, and $L-$Lipschitz derivative g'. Then there for all $\theta \in [-\pi\ \pi]$: $g(\theta) \leq g(0) + \theta g'(0) + \frac{L}{2}\theta^2$.*

Proof. We have for all θ,
$|g'(\theta) - g'(0)| \leq L|\theta|$. We now have: $g(\theta) - g(0) - \theta g'(0) = \int_0^\theta g'(\tau) - g'(0)d\tau \leq \int_0^\theta |g'(\tau) - g'(0)|d\tau \leq \int_0^\theta L|\tau|d\tau = \frac{L}{2}\theta^2$. □

Corollary 1. *Let $g = g^{t+1}_{i(t+1)j(t+1)}$. Under the conditions of Algorithm 4, we have:*
$f(U_t) - f(U_{t+1}) \geq \frac{1}{2L}\nabla_{ij} f(U_t)^2$ for the same constant L defined in 1.

Proof. By the definition of g we have $f(U_{t+1}) = \min_\theta\ g(\theta)$, and we also have $g(0) = f(U_t)$. Finally, by Eq. 1 of the main paper we have $\nabla_{ij} f(U_t) = g'(0)$. From Lemma 1, we have $g(\theta) - g(0) \leq \theta g'(0) + \frac{L}{2}\theta^2$. Minimizing the right-hand side with respect to θ, we see that $\min_\theta \{g(0) - g(\theta)\} \geq \frac{1}{2L}(g'(0))^2$. Substituting $f(U_{t+1}) = \min_\theta\ g(\theta)$,$f(U_t) = g(0)$, and $\frac{1}{2L}\nabla_{ij} f(U_t) = g'(0)$ completes the result. □

Proof of Theorem 2. By Corollary 1, we have $f(U_t) - f(U_{t+1}) \geq \frac{1}{2L}\nabla_{ij} f(U_t)^2$. Recall that $\pm\nabla_{ij} f(U_t)$ is the (i,j) and (j,i) entry of $\nabla f(U_t)$. If we take the expectation of both sides with respect to a uniform random choice of indices i, j such that $1 \leq i < j \leq d$, we have:

$$E\left[f(U_t) - f(U_{t+1})\right] \geq \frac{1}{L \cdot d^2)}||\nabla f(U_t)||^2, \tag{3}$$

Summing the left-hand side gives a telescopic sum which can be bounded by $f(U_0) - \min_{U\in\mathcal{O}_d} f(U) = f(U_0) - f_{min}$. Summing the right-hand side and using this bound, we obtain

$$\sum_{t=0}^{T} E\left[||\nabla f(U_t)||_2^2\right] \leq L \cdot d^2(f(U_0) - f_{min}) \tag{4}$$

This means that $\min_{0\leq t\leq T} E\left[||\nabla f(U_t)||_2^2\right] \leq \frac{L\cdot d^2(f(U_0)-f_{min})}{T+1}$. □

B. Proofs of theorems of Section 5

Definition 4. A tensor T is *orthogonally decomposable* if there exists an orthonormal set of vectors $v_1, \ldots v_d \in \mathbb{R}^d$, and positive scalars $\lambda_1, \ldots \lambda_d > 0$ such that:

$$T = \sum_{i=1}^{d} \lambda_i (v_i \otimes v_i \otimes v_i). \tag{5}$$

Theorem 3. Let $T \in R^{d\times d\times d}$ have an orthogonal decomposition as in Definition 4, and consider the optimization problem

$$\max_{U\in\mathcal{O}_d} f(U) = \sum_{i=1}^{d} T(u_i, u_i, u_i), \tag{6}$$

where $U = [u_1\, u_2\, \ldots\, u_d]$. The stable stationary points of the problem are exactly orthogonal matrices U such that $u_i = v_{\pi(i)}$ for a permutation π on $[d]$. The maximum value they attain is $\sum_{i=1}^d \lambda_i$.

Proof. For a tensor T' denote vec$(T') \in \mathbb{R}^{d^3}$ the vectorization of T' using some fixed order of indices. Set $\hat{T}(U) = \sum_{i=1}^d (u_i \otimes u_i \otimes u_i)$, with $\hat{T}(U)_{abc} = \sum_{i=1}^d u_{ia}u_{ib}u_{ic}$. The sum of trilinear forms in Eq. 6 is equivalent to the inner product in $\mathbb{R}^{d^3}$ between $\hat{T}(U)$ and T: $\sum_{i=1}^d T(u_i, u_i, u_i) = \sum_{i=1}^d \sum_{abc} T_{abc}u_{ia}u_{ib}u_{ic} = \sum_{abc} T_{abc}\left(\sum_{i=1}^d u_{ia}u_{ib}u_{ic}\right) = \sum_{abc} T_{abc}\hat{T}(U)_{abc} =$ vec$(T) \cdot$ vec$(\hat{T}(U))$. Consider the following two facts:
(1) $\hat{T}(U)_{abc} \leq 1\ \forall a, b, c = 1 \ldots d$: since the vectors u_i are orthogonal, all their components $u_{ia} \leq 1$. Thus $\hat{T}(U)_{abc} = \sum_{i=1}^d u_{ia}u_{ib}u_{ic} \leq \sum_{i=1}^d u_{ia}u_{ib} =\leq 1$, where the last inequality is because the sum is the inner product of two rows of an orthogonal matrix.
(2) $||$vec$(\hat{T}(U))||_2^2 = d$. This is easily checked by forming out the sum of squares explicitly, using the orthonormality of the rows and columns of the matrix U.
Assume without loss of generality that $V = I_d$. This is because we may replace the terms $T(u_i, u_i, u_i)$ in the objective with $T(V^T u_i, V^T u_i, V^T u_i)$, and because the manifold $V^T\mathcal{O}_d$ is identical to $\mathcal{O}_d$. Thus we have that T is a diagonal tensor, with $T_{aaa} = \lambda_a > 0$, $a = 1 \ldots d$. Considering facts (1) and (2) above, we have the following inequality:

$$\max_{U\in\mathcal{O}_d} \sum_{i=1}^{d} T(u_i, u_i, u_i) = \max_{U\in\mathcal{O}_d} \text{vec}(\hat{T}(U)) \cdot T \leq \tag{7}$$

$$\max_{\hat{T}} \text{vec}(\hat{T}) \cdot T \quad s.t. \quad ||\text{vec}(\hat{T})||_\infty \leq 1 \wedge ||\text{vec}(\hat{T})||_2^2 = d. \tag{8}$$

T is diagonal by assumption, with exactly d non-zero entries. Thus the maximum of (5) is attained if and only if

[2] Because $\mathcal{O}_d$ is compact, any function f with a continuous second-derivative will obey this condition.

Algorithm 5 Riemannian coordinate minimization for streaming sparse PCA

```
Input: Data stream $a_i \in \mathbb{R}^d$, number of sparse principal components $m$, initial matrix $U_0 \in \mathcal{O}_m$, sparsity parameter $\gamma \geq 0$, number of inner iterations $L$.
  $AU = [a_1 a_2 \dots a_m] \cdot U_0$ . //$AU$ is of size $d \times m$
  while not stopped do
    for $t = 1 \dots L$ do
      1. Sample uniformly at random a pair $(i(t), j(t))$ such that $1 \leq i(t) < j(t) \leq m$.
      2. $\theta_{t+1} = \underset{\theta}{\operatorname{argmax}} \sum_{k=1}^{d} ([|cos(\theta)(AU)_{ki(t)} + sin(\theta)(AU)_{kj(t)}| - \gamma]_+^2 + [| - sin(\theta)(AU)_{ki(t)} + cos(\theta)(AU)_{kj(t)}| - \gamma]_+^2)$.
      3.$AU = AU \cdot G(i(t), j(t)), \theta_{t+1})$.
    end for
    4. $i_{min} = \underset{i=1 \dots m}{\operatorname{argmin}} ||(AU)_{:,i}||_2$.
    5. Sample new data point $a_{new}$.
    6. $(AU)_{:,i_{min}} = a_{new}$.
  end while
  $Z = solveForZ(AU, \gamma)$ // Algorithm 6 of Journée et al. (2010).
Output: $Z \in \mathbb{R}^{d \times m}$
```

$\hat{T}_{aaa} = 1$, $a = 1 \dots d$, and all other entries of $\hat{T}$ are 0. The value at the maximum is then $\sum_{i=1}^{d} \lambda_i$.

The diagonal ones tensor $\hat{T}$ can be decomposed into $\sum_{i=1}^{d} e_i \otimes e_i \otimes e_i$. Interestingly, in the tensor case, unlike in the matrix case, the decomposition of orthogonal tensors is *unique* upto permutation of the factors (Kruskal, 1977; Kolda & Bader, 2009). Thus, the only solutions which attain the maximum of 7 are those where $u_i = e_{\pi(i)}$, $i = 1, \dots d$. □

C. Algorithm for streaming sparse PCA

Following are the details for the streaming sparse PCA version of our algorithm used in the experiments of section 4. The algorithm itself is brought in Algorithm 5. The algorithm starts with running the original coordinate minimization procedure on the first m samples. It then chooses the column with the least l_2 and replaces it with a new data sample, and then re-optimizes on the new set of samples. There is no need for it to converge in the inner iterations, and in practice we found that order m steps after each new sample are enough for good results.

D. Alternate version of orthogonal tensor decomposition algorithm - lazy tensor evaluation

Algorithm 3 in the main text is “Riemannian coordinate maximization for orthogonal tensor decomposition”. The version presented there assumes that the full $d \times d \times d$ tensor T is given as input to the algorithm. Typically in the applications we consider here, this tensor is formed as a third order moment from a given dataset. Let $A \in \mathbb{R}^{d \times n}$ be the data matrix, consisting of n observation with d dimensions. In the simplest case we will have that $T_{ijk} = \sum_{l=1}^{n} A_{il} A_{jl} A_{kl}$. More complex cases (for example when applying the method to fit an LDA model) still require simple vector operations which cost $O(n)$ computations to obtain each value T_{ijk}.

We can therefore adopt a lazy computation model, and refrain from constructing the entire moment tensor T in advance. Instead we may calculate the entries T_{ijk} only on demand, and on each step apply the Givens rotation to the *data matrix* instead of the tensor. This requires $O(n)$ operations, as we will be rotating the i and j dimensions (rows) of the data matrix A. See Algorithm 6 below.

Overall the computational cost of each step of this version of the algorithm is $O(n)$ where n is the number of data samples. This is compared to $O(d^2)$ operations for the version presented in the main text, where d is typically not the original data dimension, but the number of latent variables such as latent topics in LDA or mixture components in a GMM. See Anandkumar et al. (2012) for more details.

Algorithm 6 Riemannian coordinate maximization for orthogonal tensor decomposition with lazy tensor evaluation

```
Input: Data matrix $A \in \mathbb{R}^{d \times n}$. Procedure $\mathcal{S}(A)$ for obtaining single tensor entries from $A$ with computational cost $O(n)$.
  Initialize $t = 0$, $\tilde{A}_0 = A$, $U_0 = I_d$.
  while not converged do
    1. Sample uniformly at random a pair $(i(t), j(t))$ such that $1 \leq i(t) < j(t) \leq d$.
    2. Obtain $T_{iii}, T_{jjj}, T_{ijj}, T_{jii}$ from $\tilde{A}_t$ by $\mathcal{S}(\tilde{A}_t)$.
    3. $\theta_t = \underset{\theta}{\operatorname{argmax}}\, g_t^{ij}(\theta)$, where $g_t^{ij}$ is defined as in Eq. 9 of the main text.
    4. $\tilde{A}_{t+1} = G(i, j, \theta_t)^T \tilde{A}_t$.
    5. $U_{t+1} = U_t G(i, j, \theta_t)$.
    6. $t = t + 1$.
  end while
Output: $U_{final}$.
```

References

Absil, P-A, Mahony, Robert, and Sepulchre, Rodolphe. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

Anandkumar, Anima, Ge, Rong, Hsu, Daniel, Kakade, Sham M, and Telgarsky, Matus. Tensor decompositions for learning latent variable models. *arXiv preprint arXiv:1210.7559*, 2012.

Armijo, Larry. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

Bertsekas, Dimitri P. *Nonlinear programming*. Athena Scientific, 1999.

Journée, Michel, Nesterov, Yurii, Richtárik, Peter, and Sepulchre, Rodolphe. Generalized power method for sparse principal component analysis. *The Journal of Machine Learning Research*, 11:517–553, 2010.

Kolda, Tamara G and Bader, Brett W. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Kruskal, Joseph B. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18(2):95 – 138, 1977. ISSN 0024-3795. doi: 10.1016/0024-3795(77)90069-6.

Patrascu, Andrei and Necoara, Ion. Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. *arXiv preprint arXiv:1305.4027*, 2013.