Supplement to "Diagonally-Dominant Principal Component Analysis"*

Zheng Tracy Ke
Department of Statistics, Harvard University
and
Lingzhou Xue
Department of Statistics, Pennsylvania State University
and
Fan Yang
Department of Statistics, University of Chicago

September 20, 2019

In this supplementary note, we include the details about the efficient projection onto the set of "symmetric c-diagonally-dominant" matrices in Appendix A and the proposed two-block ADMM for solving the convex relaxation of Exact DD-PCA in Appendix B.

A Efficient projection onto $\mathcal{SDD}_c^+$

Recall that $\mathcal{S}$ is the set of symmetric matrices and $\mathcal{DD}_c^+$ is the set of c-diagonally-dominant matrices with nonnegative diagonal entries. Now, we present the (Euclidean) projection of a matrix $\boldsymbol{A}$ onto the convex cone $\mathcal{SDD}_c^+$ or $\mathcal{DD}_c^+$, denoted by $\mathcal{P}_{\mathcal{SDD}_c^+}(\boldsymbol{A})$ or $\mathcal{P}_{\mathcal{DD}_c^+}(\boldsymbol{A})$.

Algorithm 1. ***Mendoza-Raydan-Tarazaga (MRT) Algorithm***

Given a $p \times p$ matrix $\boldsymbol{A}$, where the jth row of $\boldsymbol{A}$ is denoted by $\mathbf{a}_j$. For $1 \le j \le p$, the jth row of the projection $\boldsymbol{X}$, denoted by $\mathbf{x}_j$, is given by

- *If $a_{jj} \ge \sum_{l:l\neq j} |a_{jl}|$, then $\mathbf{x}_j = \mathbf{a}_j$.*

*The authors gratefully acknowledge the support of NIH grant P50DA039838 and NSF grants DMS-1505256, DMS-1712958 and DMS–1811552.

- *If* $-\sum_{l:l\neq j}|a_{jl}| \leq a_{jj} < 0$ *and* $|a_{jj}| > |a_{jl}|$ *for all* $l \neq j$, *or* $a_{jj} < -\sum_{l:l\neq j}|a_{jl}|$, *then* $\mathbf{x}_j = \mathbf{0}$.

- *If* $-\sum_{l:l\neq j}|a_{jl}| \leq a_{jj} < 0$ *and* $|a_{jj}| \leq |a_{jl}|$ *for some* $l \neq j$, *or* $0 \leq a_{jj} < \sum_{l:l\neq j}|a_{jl}|$, *then* $\mathbf{x}_j$ *is generated as follows:*

  1. *Sort* $|\mathbf{a}_j|$, *excluding* a_{jj}, *in the ascending order, and denote the reordered vector as* e. *Note that* $e_j = a_{jj}$ *and* $|e_i| \leq |e_l|$ *for all* $i < l, i \neq j, l \neq j$.
  2. *For* $m \neq j$, *compute* $d_m = \sum_{l=m}^{p}|e_l| \cdot I_{\{j\neq l\}} - e_j$ *and* $\bar{d}_m = d_m/(p-m+1) \cdot I_{\{m<j\}} + d_m/(p-m+2) \cdot I_{\{m>j\}}$
  3. *Solve* $m^\star$ *as the smallest integer among* $m = 1, \ldots, p$ *such that* $m \neq j$, $|e_m| > 0$ *and* $|e_m| \geq \bar{d}_m$
  4. *Solve* $\mathbf{x}_j = (x_{j1}, \ldots, x_{jp})$ *such that* $x_{jj} = a_{jj} + \bar{d}_{m^*}$; $x_{ji} = (a_{ji} - \bar{d}_{m^*})^+$ *if* $a_{ji} \geq 0$ *for* $i \neq j$; $x_{ji} = -(a_{ji} + \bar{d}_{m^*})^-$ *if* $a_{ji} < 0$ *for* $i \neq j$, *where* $(z)^+ = \max\{z, 0\}$ *and* $(z)^- = -\min\{z, 0\}$.

Mendoza et al. (1998) applied Dykstra's alternating projection algorithm between $\mathcal{DD}^+$ and $\mathcal{S}$ to obtain the projection on $\mathcal{SDD}^+$. The algorithm is summarized in Algorithm 2.

Algorithm 2. ***Efficient Projection onto*** $\mathcal{SDD}^+$

Given a $p \times p$ *matrix* $\boldsymbol{A}$,

- *Let* $\boldsymbol{G}^{(0)} = \boldsymbol{A}$ *and* $\boldsymbol{I}^{(0)} = \mathbf{0}$
- *For* $t = 1, 2, \ldots$
  - $\boldsymbol{G}^{(t)} = \mathcal{P}_{\mathcal{DD}^+}\left(\frac{1}{2}(\boldsymbol{G}^{(t-1)} + (\boldsymbol{G}^{(t-1)})^T) - \boldsymbol{I}^{(t-1)}\right)$
  - $\boldsymbol{I}^{(t)} = \boldsymbol{G}^{(t)} - \left(\frac{1}{2}(\boldsymbol{G}^{(t-1)} + (\boldsymbol{G}^{(t-1)})^T) - \boldsymbol{I}^{(t-1)}\right)$
- *Stop if the convergence criterion is met.*

When $c = 1$, the convergence result of Algorithm 2 can be similarly established as in Boyle and Dykstra (1986) such that the iterated solutions converge in the Frobenius norm to the unique solution of the projection on $\mathcal{SDD}^+$. More details can be found in Mendoza et al. (1998). When $c \neq 1$, MRT algorithm can't be directly used. In this case, we obtain

$\mathcal{P}_{\mathcal{DD}_c^+}(\boldsymbol{A})$ through Quadratic Programming (QP). The key observation is that the problem can be separated as p independent row-wise projection. For each $1 \le j \le p$, the jth row projection can be written as

$$\min_{v_1,\dots,v_p} \sum_{i=1}^{p} (a_{ji} - v_i)^2 \qquad \text{s.t. } v_j \ge c \sum_{i:i\neq j} |v_i| \tag{1}$$

and the solution $(v_1, \dots, v_p)$ would be the jth row of $\mathcal{P}_{\mathcal{DD}_c^+}(\boldsymbol{A})$. We can reformulate (1) as

$$\min_{\delta_1,\dots,\delta_p} \sum_{i=1}^{p} \delta_i^2 \qquad \text{s.t. } a_{jj} - \delta_j \ge c \sum_{i:i\neq j} |a_{ji} - \delta_i| \tag{2}$$

It's easy to see that for $i \neq j$, we should let $\text{sign}(\delta_i) = \text{sign}(a_{ji})$ and $|\delta_i \le a_{ji}|$, and hence $|a_{ji} - \delta_i| = |a_{ji}| - |\delta_i|$. Without loss of generality, we assume $a_{ji} \ge 0$ for all $i \neq j$ so we can restrict $\delta_i \ge 0$ for all $i \neq j$. Then (2) becomes

$$\min_{\delta_1,\dots,\delta_p} \sum_{i=1}^{p} \delta_i^2 \qquad \text{s.t. } a_{jj} - \delta_j \ge c \sum_{i:i\neq j} (a_{ji} - \delta_i), \quad a_{ji} \ge \delta_i \ge 0 \text{ for all } i \neq j \tag{3}$$

which is a QP problem and can be solved using standard solver.

B Convex relaxation and ADMM for Exact DD-PCA

The exact DD-PCA is difficult to solve due to the nonconvex rank minimization. Consider the following convex relaxation of the exact DD-PCA:

$$\min_{(\boldsymbol{L},\boldsymbol{A})} \|\boldsymbol{L}\|_* \quad \text{subject to} \quad \boldsymbol{S} = \boldsymbol{L} + \boldsymbol{A}, \quad \boldsymbol{A} \in \mathcal{SDD}^+. \tag{4}$$

where $\|\cdot\|_*$ is the matrix nuclear norm.

Given the efficient projection onto $\mathcal{DD}^+$ in Algorithm 1, we introduce a new variable $\boldsymbol{B}$, satisfying the equality that $\boldsymbol{A} = \boldsymbol{B}$, to separate the symmetric and diagonally-dominant constraints as follows:

$$\min_{\boldsymbol{L},\boldsymbol{A}} \|\boldsymbol{L}\|_* + \mathcal{I}_{\boldsymbol{A}\in\mathcal{DD}^+} + \mathcal{I}_{\boldsymbol{B}=\boldsymbol{B}^T} \quad \text{subject to} \quad \boldsymbol{S} = \boldsymbol{L} + \boldsymbol{A}, \ \boldsymbol{A} - \boldsymbol{B} = 0$$

where $\mathcal{I}_C$ is the indicator function which equals to 0 if condition C is satisfied, and equals to infinity otherwise (Boyd and Vandenberghe, 2004).

We define the following augmented Lagrange function:

$$\begin{aligned}\mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{A},\boldsymbol{B},\boldsymbol{\Lambda}_1,\boldsymbol{\Lambda}_2) &= \|\boldsymbol{L}\|_* + \mathcal{I}_{\boldsymbol{A}\in\mathcal{DD}^+} + \mathcal{I}_{\boldsymbol{B}=\boldsymbol{B}^T} + \frac{\rho}{2}(\|\boldsymbol{A}-\boldsymbol{B}\|_F^2 + \|\boldsymbol{L}+\boldsymbol{A}-\boldsymbol{S}\|_F^2) \\ &\quad +\langle\boldsymbol{\Lambda}_1, \boldsymbol{A}-\boldsymbol{B}\rangle + \langle\boldsymbol{\Lambda}_2, \boldsymbol{L}+\boldsymbol{A}-\boldsymbol{S}\rangle\end{aligned}$$

where $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ are the Lagrangian multipliers associated with the equality constraints, and ρ is a given penalty parameter. We propose an efficient ADMM to solve the exact DD-PCA from $\mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{A},\boldsymbol{B},\boldsymbol{\Lambda}_1,\boldsymbol{\Lambda}_2)$, which proceeds as follows till convergence:

$$\begin{aligned}
\boldsymbol{L}\text{ step}: \quad \boldsymbol{L}^{(t)} &= \arg\min_{\boldsymbol{L}} \ \mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{A}^{(t-1)},\boldsymbol{B},\boldsymbol{\Lambda}_1^{(t-1)},\boldsymbol{\Lambda}_2^{(t-1)}) \\
\boldsymbol{B}\text{ step}: \quad \boldsymbol{B}^{(t)} &= \arg\min_{\boldsymbol{B}} \ \mathcal{L}_\rho(\boldsymbol{L},\boldsymbol{A}^{(t-1)},\boldsymbol{B},\boldsymbol{\Lambda}_1^{(t-1)},\boldsymbol{\Lambda}_2^{(t-1)}) \\
\boldsymbol{A}\text{ step}: \quad \boldsymbol{A}^{(t)} &= \arg\min_{\boldsymbol{A}} \ \mathcal{L}_\rho(\boldsymbol{L}^{(t)},\boldsymbol{A},\boldsymbol{B}^{(t)},\boldsymbol{\Lambda}_1^{(t-1)},\boldsymbol{\Lambda}_2^{(t-1)}) \\
\boldsymbol{\Lambda}_1\text{ step}: \quad \boldsymbol{\Lambda}_1^{(t)} &= \boldsymbol{\Lambda}_1^{(t-1)} + \rho(\boldsymbol{A}^{(t)} + \boldsymbol{L}^{(t)} - \boldsymbol{S}) \\
\boldsymbol{\Lambda}_2\text{ step}: \quad \boldsymbol{\Lambda}_2^{(t)} &= \boldsymbol{\Lambda}_2^{(t-1)} + \rho(\boldsymbol{A}^{(t)} - \boldsymbol{B}^{(t)})
\end{aligned}$$

Our proposed ADMM is a two-block ADMM with two blocks $\{\boldsymbol{L},\boldsymbol{B}\}$ and $\boldsymbol{A}$, and its global convergence is always guaranteed (Boyd et al., 2011). In what follows, we explicitly show how to obtain closed-form solutions for each subproblem. In the $\boldsymbol{L}$ step, we have

$$\begin{aligned}
\boldsymbol{L}^{(t)} &= \arg\min_{\boldsymbol{L}} \ \|\boldsymbol{L}\|_* + \frac{\rho}{2}\|\boldsymbol{A}^{(t-1)} + \boldsymbol{L} - \boldsymbol{S}\|_F^2 + \langle\boldsymbol{\Lambda}_1^{(t-1)}, \boldsymbol{A}^{(t-1)} + \boldsymbol{L} - \boldsymbol{S}\rangle \\
&= \arg\min_{\boldsymbol{L}} \ \frac{1}{2}\|\boldsymbol{L} + \boldsymbol{A}^{(t-1)} - \boldsymbol{S} + \rho^{-1}\boldsymbol{\Lambda}_1^{(t-1)}\|_F^2 + \rho^{-1}\|\boldsymbol{L}\|_*
\end{aligned}$$

It's easy to show that the solution is given by $\boldsymbol{L}^{(t)} = \mathcal{D}_{\rho^{-1}}\left(\boldsymbol{S} - \boldsymbol{A}^{(t-1)} - \rho^{-1}\boldsymbol{\Lambda}_1^{(t-1)}\right)$ where $\mathcal{D}_\tau(\boldsymbol{\Omega})$ is the singular value thresholding operator given by $\mathcal{D}_\tau(\boldsymbol{\Omega}) = \boldsymbol{U}s_\tau(\boldsymbol{D})\boldsymbol{V}^T$ for any singular value decomposition $\boldsymbol{\Omega} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, and s_τ denotes the soft-thresholding operator given by $s_\tau(x) = \mathrm{sgn}(x)\max(|x| - \tau, 0)$.

In the $\boldsymbol{B}$ step, we also have the following closed-form solution:

$$\begin{aligned}
\boldsymbol{B}^{(t)} &= \arg\min_{\boldsymbol{B}} \ \mathcal{I}_{\boldsymbol{B}=\boldsymbol{B}^T} + \frac{\rho}{2}\|\boldsymbol{A}^{(t-1)} - \boldsymbol{B}\|_F^2 + \langle\boldsymbol{\Lambda}_2^{(t-1)}, \boldsymbol{A}^{(t-1)} - \boldsymbol{B}\rangle \\
&= \arg\min_{\boldsymbol{B}} \ \mathcal{I}_{\boldsymbol{B}=\boldsymbol{B}^T} + \frac{\rho}{2}\|\rho^{-1}\boldsymbol{\Lambda}_2^{(t-1)} + \boldsymbol{A}^{(t-1)} - \boldsymbol{B}\|_F^2 \\
&= \frac{1}{2}\left[\left(\boldsymbol{A}^{(t-1)} + \rho^{-1}\boldsymbol{\Lambda}_2^{(t-1)}\right) + \left(\boldsymbol{A}^{(t-1)} + \rho^{-1}\boldsymbol{\Lambda}_2^{(t-1)}\right)^T\right]
\end{aligned}$$

Finally in the $\boldsymbol{A}$ step, we have

$$\begin{aligned}
\boldsymbol{A}^{(t)} &= \arg\min_{\boldsymbol{A}} \ \mathcal{I}_{\boldsymbol{A}\in\mathcal{DD}^+} + \frac{\rho}{2}\left(\|\boldsymbol{A}+\boldsymbol{L}^{(t)}-\boldsymbol{S}+\rho^{-1}\boldsymbol{\Lambda}_1^{(t-1)}\|_F^2 + \|\boldsymbol{A}-\boldsymbol{B}^{(t)}+\rho^{-1}\boldsymbol{\Lambda}_2^{(t-1)}\|_F^2\right) \\
&= \arg\min_{\boldsymbol{A}} \ \mathcal{I}_{\boldsymbol{A}\in\mathcal{DD}^+} + \rho\left\|\boldsymbol{A}+\frac{1}{2}\left(\boldsymbol{L}^{(t)}-\boldsymbol{S}+\rho^{-1}\boldsymbol{\Lambda}_1-\boldsymbol{B}^{(t)}+\rho^{-1}\boldsymbol{\Lambda}_2^{(t-1)}\right)\right\|_F^2 \\
&= \mathcal{P}_{\mathcal{DD}^+}\left(\frac{1}{2}(\boldsymbol{S}-\boldsymbol{L}^{(t)}+\boldsymbol{B}^{(t)}-\rho^{-1}\boldsymbol{\Lambda}_1^{(t-1)}-\rho^{-1}\boldsymbol{\Lambda}_2^{(t-1)})\right)
\end{aligned}$$

We summarize our proposed two-block ADMM in Algorithm 3.

Algorithm 3. ***Two-Block ADMM for Solving the Exact DD-PCA***

Given the sample covariance matrix $\boldsymbol{S}$, *do*

- *Let* $\boldsymbol{A}^{(0)} = \boldsymbol{\Lambda}_1^{(0)} = \boldsymbol{\Lambda}_2^{(0)} = \boldsymbol{0}$
- *For* $t = 1, 2, \ldots$
  - $\boldsymbol{L}^{(t)} = \mathcal{D}_{\rho^{-1}}\left(\boldsymbol{S}-\boldsymbol{A}^{(t-1)}-\rho^{-1}\boldsymbol{\Lambda}_1^{(t-1)}\right)$.
  - $\boldsymbol{B}^{(t)} = \frac{1}{2}\left[\left(\boldsymbol{A}^{(t-1)}+\rho^{-1}\boldsymbol{\Lambda}_2^{(t-1)}\right)+\left(\boldsymbol{A}^{(t-1)}+\rho^{-1}\boldsymbol{\Lambda}_2^{(t-1)}\right)^T\right]$
  - $\boldsymbol{A}^{(t)} = \mathcal{P}_{\mathcal{DD}^+}\left(\frac{1}{2}(\boldsymbol{S}-\boldsymbol{L}^{(t)}+\boldsymbol{B}^{(t)}-\rho^{-1}\boldsymbol{\Lambda}_1^{(t-1)}-\rho^{-1}\boldsymbol{\Lambda}_2^{(t-1)})\right)$
  - $\boldsymbol{\Lambda}_1^{(t)} = \boldsymbol{\Lambda}_1^{(t-1)}+\rho(\boldsymbol{A}^{(t)}+\boldsymbol{L}^{(t)}-\boldsymbol{S})$
  - $\boldsymbol{\Lambda}_2^{(t)} = \boldsymbol{\Lambda}_2^{(t-1)}+\rho(\boldsymbol{A}^{(t)}-\boldsymbol{B}^{(t)})$
- *Stop if the convergence criterion is met.*

References

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011), "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, 3, 1–122.

Boyd, S. and Vandenberghe, L. (2004), *Convex Optimization*, Cambridge University Press.

Boyle, J. P. and Dykstra, R. L. (1986), "A method for finding projections onto the intersection of convex sets in Hilbert spaces," in *Advances in Order Restricted Statistical Inference*, Springer, pp. 28–47.

Mendoza, M., Raydan, M., and Tarazaga, P. (1998), "Computing the nearest diagonally dominant matrix," *Numerical Linear Algebra with Applications*, 5, 461–474.