Nonconvex ADMM for Distributed Sparse PCA

Davood Hajienzhad
Joint work Mingyi Hong

Iowa State University

Presented at GlobalSIP 2015

The Main Contribution

- **Question:** How to perform principal component analysis over a massively distributed data set?

- **Our contribution:** Design and analysis an efficient nonconvex algorithm.

Outline

Principal Component Analysis(PCA)

- **PCA** aims to reduce the dimension of multi-variate data set.
- For given data set D, the solution of:

$$\max_{x} \ \|Dx\|_2^2, \quad \text{s.t. } \|x\|_2^2 \leq 1 \tag{1}$$

is called first loading vector and the vector Dx is called the first PC [Mackey (2008)] .

- $\|Dx\|_2^2$ represents the explained variance of the first PC.

Sparse PCA

- **Deficiency of PCA:** Most of the PCs' coefficients are non-zero, making the resulting solutions difficult to interpret.

- **How to address this issue?** Using Sparse PCA (SPCA):

$$\max_{x} \; \|Dx\|_2^2 - \lambda r(x), \quad \text{s.t. } \|x\|_2^2 \leq 1 \tag{2}$$

where $r(x)$ is a sparsity-promoting, and $\lambda > 0$ controlling the sparsity. [Kwak (2008)].

- $r(x)$ can be : $\|x\|_0$, or its approximations such as $\|x\|_1$ (convex), $\sum_i \log(\epsilon + |x_i|)$ (non-convex).

Literature in SPCA

- [D'Aspremont *et al* (2007)]: Proposed a semi-definite relaxation of a rank constrained problem (DSPCA).
- [Shen et al (2008)] : Used the connection of PCA with SVD and solved a low rank matrix approximation to extract the PCs (sPCA-rSVD).
- [Journee et al (2010)]: Formulated SPCA as maximization of a convex function on a compact set (G-Power).
- [Zhao et al (2015)]: Proposed a block coordinate descent (BCD) method for solving SPCA (BCD-SPCA).

Benefit of Distributed Computing

- **Question**: **Why we need distributed optimization?**

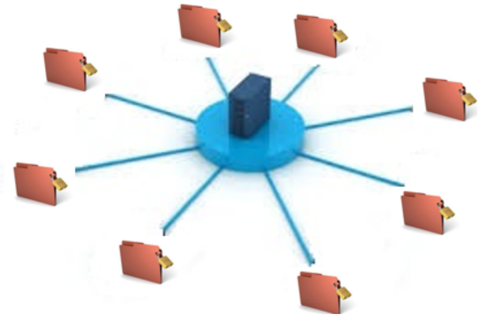
  (1) Data are collected/stored in a distributed network.

Benefit of Distributed Computing

(2) Memory Limitation

Benefit of Distributed Computing

(3) Privacy Issue

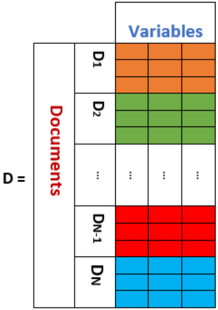


Benefit of Distributed Computing

(4) Parallel Clusters

Distribution Across the Rows

- Splitting the rows of $D \in \mathbb{R}^{n \times p}$ into N sub-matrix:

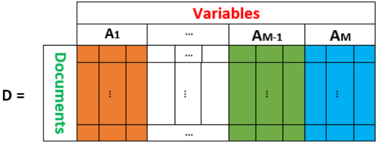


- SPCA problem can be reformulated:

$$\max_{x} \sum_{i=1}^{N} \|D_i x\|_2^2 - \lambda r(x), \quad \text{s.t. } \|x\|_2^2 \leq 1. \tag{3}$$

Distribution Across the Columns

- Splitting the columns of $D \in \mathbb{R}^{n \times p}$ into M sub-matrix:



- SPCA problem can be reformulated:

$$\max \left\| \sum_{i=1}^{M} A_i x_i \right\|^2 - \lambda r(x), \quad \text{s.t.} \quad \|x\|_2^2 \leq 1, \qquad (4)$$

- Both formulations are non-convex optimization problem.

1. Introduction
2. Distributed SPCA Formulations
3. Proposed ADMM Algorithm
4. Numerical Results
   - Performance on Centralized Data
   - Performance on Distributed Data

ADMM setup when rows are distributed

- Define new variable z:

$$\begin{aligned} \min_{x,z} \quad & \textstyle\sum_{i=1}^{N} -\|D_i x_i\|_2^2 + \lambda r(z) \\ \text{s.t.} \quad & \|z\| \leq 1,\ x_i = z,\ i = 1, \cdots N; \end{aligned} \tag{5}$$

- Hong *et al.*(2014) showed that the ADMM converges to the set of stationary solutions when $r(x)$ is convex.

- In our case $r(z)$ is also allowed to be non-convex

ADMM setup when rows are distributed

- Augmented Lagrangian function

$$L_\rho(x, z; y) = -\sum_{i=1}^{N} \|D_i x_i\|_2^2 + \lambda r(z) + \sum_{i=1}^{N} \langle x_i - z, y_i \rangle + \sum_{i=1}^{N} \frac{\rho_i}{2} \|x_i - z\|^2$$

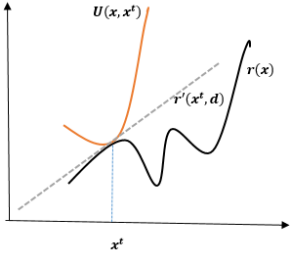
  $y := \{y_i \in \mathbb{R}^p\}_{i=1}^N$ is the set of dual variables; $\rho_i > 0$ is penalization parameter.

- **ADMM Algorithm**: First, minimizing $L_\rho(\cdot)$ with respect to z, then with respect to $\{x_i\}$, followed by an approximate dual ascent update for $\{y_i\}$ [Boyd et al (2011)].

Non-Convex Regulizer

- **How to deal with non-convex regulizer?** Applying convex approximation technique called the block successive upper-bound minimization (BSUM) [Razaviyayn-Hong-Luo 2013].

- At iteration t, regularizer $r(z)$ is replaced with a convex upper-bound approximation, $u(z, v)$ such that:
  1. $u(v, v) = r(v)$
  2. $u'(z, v; d)|_{z=v} = r'(v; d)$
  3. $u(z, v) \geq r(v)$, for all $z, v \in X$.
  4. $u(z, v)$ is continuous $\forall z, v \in X$.



Non-Convex Regulizer

- For example, upper-bounds for the LSP and M-LSP:

1. The nonconvex LSP, $r(x) = \sum_{j=1}^{p} \log(\epsilon_j + |x_j|)$.
2. The modified LSP (M-LSP), $r(x) = \log(\epsilon + \|x\|_1)$.

$$u(x, x^t) = \begin{cases} \sum_{j=1}^{p} \frac{1}{\epsilon_j + |x_j^t|} \left(|x_j| - |x_j^t| \right) & \text{(LSP)} \\ \frac{1}{\epsilon + \|x^t\|_1} \left(\|x\|_1 - \|x^t\|_1 \right) & \text{(M-LSP)} \end{cases}.$$

ADMM algorithm when rows are distributed

Algorithm 1. ADMM for SPCA

Distribute the data into to different nodes.

Initialize the variables.

At iteration $t+1$, do:

S1: The **central node** updates z:

$$z^{t+1} = \arg\min_{\|z\|_2^2 \leq 1} \lambda u(z, z^t) + \sum_{i=1}^{N} \rho_i/2 \|x_i^t - z + y_i^t/\rho_i\|^2.$$

S2: Each node i updates x_i in **parallel**:

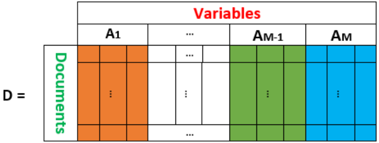
$$x_i^{t+1} = \arg\min_{x_i} -\|D_i x_i\|_2^2 + \rho_i/2\|x_i - z^{t+1} + y_i^t/\rho_i\|^2.$$

S3: Each node i updates the dual variables in **parallel**:

$$y_i^{t+1} = y_i^t + \rho_i(x_i^{t+1} - z^{t+1}).$$

ADMM setup when columns are distributed

- Splitting the columns:



- Introducing set of variables $\{z_i\}$

$$\begin{array}{ll} \min & -\left\|\sum_{i=1}^{M} z_i\right\|^2 + \lambda r(x) \\ \text{s.t.} & \|x\|^2 \leq 1, \quad A_i x_i = z_i, \quad i = 1, 2, \cdots M. \end{array}$$

- Augmented Lagrangian:

$$L_\beta(x, z; y) = -\|\sum_{i=1}^{M} z_i\|_2^2 + \lambda r(x) + \sum_{i=1}^{M} \frac{\beta_i}{2} \|A_i x_i - z_i - y_i/\beta_i\|^2.$$

ADMM algorithm when columns are distributed

Distribute the data A_i's to different nodes.

At iteration $t+1$

S1: Each node i updates x_i in **parallel**:

$$\widetilde{x}_i^{t+1} = \arg\min_{x_i} \lambda u_i(x_i, x_i^r) + \frac{L_i\beta_i}{2}\|x_i - x_i^t\|^2 + \beta_i\langle A_i^T(A_i x_i^t - z_i^t + y_i^t/\beta_i), x_i - x_i^t\rangle$$

S2: Each node sends $c_i^{t+1} = \|\widetilde{x}_i^{t+1}\|_2^2$ to the central node.

S3: Central node broadcasts $c^{t+1} = \max\{\sum_{i=1}^M c_i^{t+1}, 1\}$.

S4: Each node computes in **parallel**: $x_i^{t+1} = \widetilde{x}_i^{t+1}/\sqrt{c^{t+1}}$.

S5: The central node updates z:

$$z^{t+1} = \arg\min_{z} \; -\|\sum_{i=1}^M z_i\|_2^2 + \sum_{i=1}^M \beta_i/2\|A_i x_i^{t+1} - z_i + y_i^t/\beta_i\|^2.$$

S6: Each node i updates the dual variables in **parallel**:

$$y_i^{t+1} = y_i^t + \beta_i(A_i x_i^{t+1} - z_i^{t+1}).$$

Convergence Analysis

Theorem

We have the following convergence result for Algorithm 1-2:

(1) For Algorithm 1: *If* $\rho_i \geq 4\|D_i^\top D_i\|_2$ *for all* i*, then we have:*

$$\lim_{t\to\infty} \|x_i^{t+1} - z^{t+1}\| = 0, \ i = 1, \cdots, N.$$

Further, the algorithm converges to the set of **stationary solutions** *of SPCA.*

(2) For Algorithm 2: *If* $\beta_i \geq 4M$ *for all* i*, then we have:*

$$\lim_{t\to\infty} \|A_i x_i^{t+1} - z_i^{t+1}\| = 0, \ i = 1, \cdots, M.$$

Further, the algorithm converges to the set of **stationary solutions** *of SPCA.*

Numerical Results on Pitprops data set

- Centralized version of algorithm ($N = M = 1$).
- Pitprops data consists of 180 observations and 13 variables.

Method	Cardinality	EV
DSPCA [d'Aspremont et al (2007)]	18	79.18
sPCA-rSVD$_{\ell_0}$ [Shen et al (2008)]	18	80.85
sPCA-rSVD$_{\ell_1}$ [Shen et al (2008)]	18	80.40
Gpower$_{\ell_0}$ [Journee et al (2010)]	18	80.64
Gpower$_{\ell_1}$ [Journee et al (2010)]	19	81.11
BCD-SPCA$_{\ell_0}$ [Zhao et al (2015)]	18	80.47
BCD-SPCA$_{\ell_1}$ [Zhao et al (2015)]	18	81.14
ADMM$_{\ell_1}$ [Our Method]	18	82.93
ADMM$_{\mathrm{MLSP}}$ [Our Method]	18	**83.48**

Splitting The Rows

- We set $n = 1,000,000,\ p = 2000$.
- Randomly generated sparse matrix (95% of elements are zero), a randomly generated dense matrix.
- We split this matrix across the rows into $N \in \{16, 32, 64\}$ subsets.
- The explained variances in all cases are about 0.064.

	Cardinality		Time (Sec)		Iteration	
N	Sparse	Dense	Sparse	Dense	Sparse	Dense
16	1585	1580	40.1	45.3	2000	2250
32	1574	1574	43.9	117.5	2144	3150
64	1585	1572	110.1	397.7	2489	3868

Splitting The Columns

- Set $n = 2000$ and $p = 100,000$.
- Let $M \in \{1, 2, 4, 8, 16, 32, 64\}$.
- Apply Algorithm 2, using the M-LSP regularizer.

	Cardinality		Time (Sec)		Iteration	
M	Sparse	Dense	Sparse	Dense	Sparse	Dense
1	11960	11965	59.90	249.09	58	208
2	11960	11964	43.22	121.19	88	259
4	11962	11965	40.19	80.39	168	321
8	11963	11963	30.58	54.77	222	392
16	11962	11965	23.90	41.61	290	469
32	11962	11964	13.85	25.22	328	548
64	11961	11964	19.75	31.98	448	611

Conclusion

- We propose non-convex ADMM algorithms to solve **distributed SPCA** problems.
- Data matrix can be distributed across the **rows** as well as **columns**.
- Our methods deal with **non-convex regulizers** to promote sparsity.

Future Works

- Extend the **star network** to an arbitrary one with non-convex functions.
- Try to find conditions under which we can reach the **global optimal** solution.
- Apply the same way to prove the convergence of ADMM for more non-convex cases.

Thanks for Your Attention.