Conditional Gradient Algorithms for Rank-One Matrix Approximations with a Sparsity Constraint

Marc Teboulle

School of Mathematical Sciences
Tel Aviv University

Joint work with Ronny Luss

Optimization and Statistical Learning – OSL 2013
January 6–11, 2013 – Les Houches, France

Sparsity Constrained Rank-One Matrix Approximation $\equiv$ PCA

Principal Component Analysis solves

$$\min\{\|A - xx^T\|_F^2 : \|x\|_2 = 1,\ x \in \mathbf{R}^n\} \ \Leftrightarrow\ \max\{x^T A x : \|x\|_2 = 1,\ x \in \mathbf{R}^n\},\ (A \in \mathbb{S}_+^n)$$

Sparsity Constrained Rank-One Matrix Approximation $\equiv$ PCA

Principal Component Analysis solves

$$\min\{\|A - xx^T\|_F^2 : \|x\|_2 = 1,\ x \in \mathbf{R}^n\} \ \Leftrightarrow\ \max\{x^T A x : \|x\|_2 = 1,\ x \in \mathbf{R}^n\},\ (A \in \mathbb{S}_+^n)$$

Sparse Principal Component Analysis solves

$$\max\{x^T A x : \|x\|_2 = 1,\ \|x\|_0 \le k,\ x \in \mathbf{R}^n\},\ k \in [1, n] \text{ sparsity}$$

$\|x\|_0$ counts the number of nonzero entries of x

Sparsity Constrained Rank-One Matrix Approximation $\equiv$ PCA

Principal Component Analysis solves

$$\min\{\|A - xx^T\|_F^2 : \|x\|_2 = 1,\ x \in \mathbf{R}^n\} \Leftrightarrow \max\{x^T A x : \|x\|_2 = 1,\ x \in \mathbf{R}^n\},\ (A \in \mathbb{S}_+^n)$$

Sparse Principal Component Analysis solves

$$\max\{x^T A x : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\},\ k \in [1, n] \text{ sparsity}$$

$\|x\|_0$ counts the number of nonzero entries of x

Difficulties:

1. Maximizing a *Convex* objective.
2. Hard Nonconvex Constraint $\|x\|_0 \leq k$.

Current Approaches:

1. SDP Convex Relaxations [D'aspremont-El Ghaoui-Jordan-Lankcriet 07]
2. Approximation/Modified formulations [Many....]

Sparse PCA via Penalization/Relaxation/Approximation

The problem of interest is the difficult sparse PCA problem as is

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\}$$

Literature has focused on solving various modifications:

Sparse PCA via Penalization/Relaxation/Approximation

The problem of interest is the difficult sparse PCA problem as is

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\}$$

Literature has focused on solving various modifications:

- **l_0-penalized PCA**

$$\max\left\{x^T Ax - s\|x\|_0 : \|x\|_2 = 1\right\},\ s > 0$$

Sparse PCA via Penalization/Relaxation/Approximation

The problem of interest is the difficult sparse PCA problem as is

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\}$$

Literature has focused on solving various modifications:

- **l_0-penalized PCA**

$$\max\{x^T Ax - s\|x\|_0 : \|x\|_2 = 1\},\ s > 0$$

- **Relaxed l_1-constrained PCA** $(\|x\|_1 \leq \sqrt{\|x\|_0}\|x\|_2,\ \forall x)$

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_1 \leq \sqrt{k}\}$$

Sparse PCA via Penalization/Relaxation/Approximation

The problem of interest is the difficult sparse PCA problem as is

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\}$$

Literature has focused on solving various modifications:

- **l_0-penalized PCA**

$$\max\left\{x^T Ax - s\|x\|_0 : \|x\|_2 = 1\right\},\ s > 0$$

- **Relaxed l_1-constrained PCA** ($\|x\|_1 \leq \sqrt{\|x\|_0}\|x\|_2,\ \forall x$)

$$\max\left\{x^T Ax : \|x\|_2 = 1,\ \|x\|_1 \leq \sqrt{k}\right\}$$

- **Relaxed l_1-penalized PCA**

$$\max\left\{x^T Ax - s\|x\|_1 : \|x\|_2 = 1\right\}$$

Sparse PCA via Penalization/Relaxation/Approximation

The problem of interest is the difficult sparse PCA problem as is

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \le k,\ x \in \mathbf{R}^n\}$$

Literature has focused on solving various modifications:

- **l_0-penalized PCA**

$$\max\left\{x^T Ax - s\|x\|_0 : \|x\|_2 = 1\right\},\ s > 0$$

- **Relaxed l_1-constrained PCA** ($\|x\|_1 \le \sqrt{\|x\|_0}\|x\|_2,\ \forall x$)

$$\max\left\{x^T Ax : \|x\|_2 = 1,\ \|x\|_1 \le \sqrt{k}\right\}$$

- **Relaxed l_1-penalized PCA**

$$\max\left\{x^T Ax - s\|x\|_1 : \|x\|_2 = 1\right\}$$

- **Approximate-Penalized:** Uses concave approximation of $\|x\|_0$

$$\max\left\{x^T Ax - s\varphi_p(|x|) : \|x\|_2 = 1\right\} \ \varphi_p(x) \simeq \|x\|_0,\ p \to 0^+.$$

Sparse PCA via Penalization/Relaxation/Approximation

The problem of interest is the difficult sparse PCA problem as is

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \le k,\ x \in \mathbf{R}^n\}$$

Literature has focused on solving various modifications:

- **l_0-penalized PCA**

$$\max\{x^T Ax - s\|x\|_0 : \|x\|_2 = 1\},\ s > 0$$

- **Relaxed l_1-constrained PCA** $(\|x\|_1 \le \sqrt{\|x\|_0}\|x\|_2,\ \forall x)$

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_1 \le \sqrt{k}\}$$

- **Relaxed l_1-penalized PCA**

$$\max\{x^T Ax - s\|x\|_1 : \|x\|_2 = 1\}$$

- **Approximate-Penalized:** Uses concave approximation of $\|x\|_0$

$$\max\{x^T Ax - s\varphi_p(|x\|) : \|x\|_2 = 1\}\ \varphi_p(x) \simeq \|x\|_0,\ p \to 0^+.$$

- **SDP-Convex Relaxation** $\max\{\mathrm{tr}(AX) :\ \mathrm{tr}\,(X) = 1, X \succeq 0, \|X\|_1 \le k\}$

Convex relaxations can be computationally expensive for very large problems and will not be discussed here.

Quick Highlight of Simple Algorithms on "Modified Problems"

Type	Iteration	Per-Iteration Complexity	References
l_1-constrained	$x_i^{j+1} = \frac{\text{sgn}(((A+\frac{\sigma}{2})x^j)_i)(\lvert((A+\frac{\sigma}{2})x^j)_i\rvert-\lambda^j)_+}{\sqrt{\sum_h (\lvert((A+\frac{\sigma}{2})x^j)_h\rvert-\lambda^j)_+^2}}$	$O(n^2)$, $O(mn)$	Witten et al. (2009)
l_1-constrained	$x_i^{j+1} = \frac{\text{sgn}((Ax^j)_i)(\lvert(Ax^j)_i\rvert-s^j)_+}{\sqrt{\sum_h (\lvert(Ax^j)_h\rvert-s^j)_+^2}}$ where s^j is $(k+1)$-largest entry of vector $\lvert Ax^j\rvert$	$O(n^2)$, $O(mn)$	Sigg-Buhman (2008)
l_0-penalized	$z^{j+1} = \frac{\sum_i [\text{sgn}((b_i^T z^j)^2-s)]_+(b_i^T z^j)b_i}{\lVert \sum_i [\text{sgn}((b_i^T z^j)^2-s)]_+(b_i^T z^j)b_i \rVert_2}$	$O(mn)$	Shen-Huang (2008), Journee et al. (2010)
l_0-penalized	$x_i^{j+1} = \frac{\text{sgn}(2(Ax^j)_i)(\lvert 2(Ax^j)_i\rvert - s\varphi_p'(\lvert x_i^j\rvert))_+}{\sqrt{\sum_h (\lvert 2(Ax^j)_h\rvert - s\varphi_p'(\lvert x_h^j\rvert))_+^2}}$	$O(n^2)$	Sriperumbudur et al. (2010)
l_1-penalized	$y^{j+1} = \underset{y}{\text{argmin}} \left\{ \sum_i \lVert b_i - x^j y^T b_i \rVert_2^2 + \lambda \lVert y \rVert_2^2 + s \lVert y \rVert_1 \right\}$ $x^{j+1} = \frac{(\sum_i b_i b_i^T) y^{j+1}}{\lVert (\sum_i b_i b_i^T) y^{j+1} \rVert_2}$		Zou et al. (2006)
l_1-penalized	$z^{j+1} = \frac{\sum_i (\lvert b_i^T z^j\rvert - s)_+ \text{sgn}(b_i^T z^j) b_i}{\lVert \sum_i (\lvert b_i^T z^j\rvert - s)_+ \text{sgn}(b_i^T z^j) b_i \rVert_2}$	$O(mn)$	Shen-Huang (2008), Journee et al. (2010)

Table : Cheap sparse PCA algorithms for modified problems.

A Plethora of Models/Algorithms Revisited

All previous listed algorithms have been derived from various disparate approaches/motivations to solve **modifications** of SPCA:

- Nonsmooth reformulations
- Expectation Maximization
- Majoration-Mininimization techniques
- DC programming
- ... etc...

Q1: Are all these algorithms different? ...Any connection?

A Plethora of Models/Algorithms Revisited

All previous listed algorithms have been derived from various disparate approaches/motivations to solve **modifications** of SPCA:

- Nonsmooth reformulations
- Expectation Maximization
- Majoration-Mininimization techniques
- DC programming
- ... etc...

Q1: Are all these algorithms different? ...Any connection?

Our problem of interest is the difficult sparse PCA problem "as is"

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \le k,\ x \in \mathbf{R}^n\}$$

Q2: Is is possible to derive a simple/cheap scheme to tackle directly the sparse PCA problem as is?

A Plethora of Models/Algorithms Revisited

All previous listed algorithms have been derived from various disparate approaches/motivations to solve **modifications** of SPCA:

- Nonsmooth reformulations
- Expectation Maximization
- Majoration-Mininimization techniques
- DC programming
- ... etc...

Q1: Are all these algorithms different? ...Any connection?

Our problem of interest is the difficult sparse PCA problem "as is"

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\}$$

Q2: Is is possible to derive a simple/cheap scheme to tackle directly the sparse PCA problem as is?

Answers

- All the previously listed algorithms are a particular realization of a **"Father Algorithm": ConGradU**
  (based on the well-known Conditional Gradient Algorithm)
- **ConGradU CAN be applied directly to the original problem!**

The Conditional Gradient/Frank-Wolfe Algorithm

[Frank-Wolfe'56, Rubinov'64, Levitin-Polyak'66, Canon-Cullum' 68, Dunn'79,....]

♣ **Classic Conditional Gradient Algorithm** solves

$$\max \{F(x) : x \in C\}$$

- $F : \mathbf{R}^n \to \mathbf{R}$ is continuously differentiable
- C is nonempty, **convex** compact subset of $\mathbb{R}^n$

via the following iteration for all $j \geq 0$:

$$x^0 \in C, \; x^{j+1} = x^j + \alpha^j (p^j - x^j)$$

with

$$p^j = \operatorname{argmax} \{ \langle x - x^j, \nabla F(x^j) \rangle : x \in C \}$$

where $\alpha^j \in (0, 1]$ is a stepsize (exact/or via line search).

♠ **Here in SPCA :**

F is convex, possibly nonsmooth; (through equiv. reformulations)

C is compact but *nonconvex*

Maximizing a Convex function over a Compact Nonconvex set

ConGradU – Conditional Gradient with a Unit Step Size

$$x^0 \in C, \; x^{j+1} \in \operatorname{argmax}\{\langle x - x^j, F'(x^j)\rangle : x \in C\}$$

Notes:

1. Mangasarian (96) considered it for C a polyhedral set.
2. F is not assumed to be differentiable and $F'(x)$ is a subgradient of F at x.
3. The algorithm is useful when $\max\{\langle x - x^j, F'(x^j)\rangle : x \in C\}$ is simple to solve

Maximizing a Convex function over a Compact Nonconvex set

ConGradU – Conditional Gradient with a Unit Step Size

$$x^0 \in C, \; x^{j+1} \in \operatorname{argmax}\{\langle x - x^j, F'(x^j)\rangle : x \in C\}$$

Notes:

1. Mangasarian (96) considered it for C a polyhedral set.
2. F is not assumed to be differentiable and $F'(x)$ is a subgradient of F at x.
3. The algorithm is useful when $\max\{\langle x - x^j, F'(x^j)\rangle : x \in C\}$ is simple to solve

A Basic Convergence Result

(a) The sequence $F(x^j)$ is monotonically increasing and

$$\lim_{j \to \infty} \gamma(x^j) = 0, \text{ where } \gamma(x) := \max\{\langle u - x, F'(x)\rangle : \; u \in C\}.$$

(b) If F is assumed continuously differentiable, then every limit point of the sequence $\{x^j\}$ converges to a stationary point.

The Original l_0-constrained PCA via ConGradU

Applying **ConGradU** directly to

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\}$$

results in the iteration

$$x^{j+1} = \operatorname{argmax}\{x^{jT} Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k\},\ j = 0, 1, \ldots$$

The Original l_0-constrained PCA via ConGradU

Applying **ConGradU** directly to

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\}$$

results in the iteration

$$x^{j+1} = \operatorname{argmax}\{x^{jT} Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k\},\ j = 0, 1, \ldots$$

- Thus, the main step consists of maximizing *a linear function* on intersection of two nonconvex sets

$$x \in C_1 \cap C_2 \text{ with } C_1 := \{x :\ \|x\|_2 = 1\},\ C_2 := \{x :\ \|x\|_0 \leq k\}$$

- It turns out that this problem is very simple!
- In fact, thanks to C_1: $x^{j+1} = \underset{x \in C_1 \cap C_2}{\operatorname{argmin}} \|x - A^T x^j\|^2 = P_{C_1 \cap C_2}(A^T x^j)$...and...

The Original l_0-constrained PCA via ConGradU

Applying **ConGradU** directly to

$$\max\{x^T Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k,\ x \in \mathbf{R}^n\}$$

results in the iteration

$$x^{j+1} = \text{argmax}\{x^{jT} Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k\},\ j = 0, 1, \ldots$$

- Thus, the main step consists of maximizing *a linear function* on intersection of two nonconvex sets

$$x \in C_1 \cap C_2 \text{ with } C_1 := \{x :\ \|x\|_2 = 1\},\ C_2 := \{x :\ \|x\|_0 \leq k\}$$

- It turns out that this problem is very simple!
- In fact, thanks to C_1: $x^{j+1} = \underset{x \in C_1 \cap C_2}{\text{argmin}} \|x - A^T x^j\|^2 = P_{C_1 \cap C_2}(A^T x^j)$...and...
- Thanks to the "hard" constraint C_2...Projection on intersection "easy"...!

$$P_{C_1 \cap C_2}(A^T x^j) \equiv P_{C_1} \circ [P_{C_2}(A^T x^j)]$$

A Simple Key Result

A Simple Key Result Given $0 \neq a \in \mathbf{R}^n$,

$$\max_x \{a^T x : \|x\|_2 = 1,\ \|x\|_0 \leq k\} = \|T_k(a)\|_2, \text{ with solution } x^* = \frac{T_k(a)}{\|T_k(a)\|_2}$$

$$(T_k(a))_i = \begin{cases} a_i, & \text{for } k \text{ largest entries (in absolute values) of } a; \\ 0, & \text{otherwise.} \end{cases}$$

A Simple Key Result

A Simple Key Result Given $0 \neq a \in \mathbf{R}^n$,

$$\max_x \{a^T x : \|x\|_2 = 1, \ \|x\|_0 \leq k\} = \|T_k(a)\|_2, \text{ with solution } x^* = \frac{T_k(a)}{\|T_k(a)\|_2}$$

$$(T_k(a))_i = \begin{cases} a_i, & \text{for } k \text{ largest entries (in absolute values) of } a; \\ 0, & \text{otherwise.} \end{cases}$$

Definition $T_k : \mathbf{R}^n \to \mathbf{R}^n$ is the best k-sparse approximation of a

$$T_k(a) := \operatorname*{argmin}_x \{\|x - a\|_2^2 : \|x\|_0 \leq k\}$$

Despite the nonconvex constraint, very easy to compute. In case k largest entries are not uniquely defined, we select the smallest possible indices, with w.l.o.g, $a \in \mathbf{R}^n$ such $|a_1| \geq \ldots \geq |a_n|$.

Computing $T_k(\cdot)$ only requires determining the k^{th} largest number of a vector of n numbers which can be done in $O(n)$ time (Blum 73) and zeroing out the proper components in one more pass of the n numbers.

l_0-constrained PCA via ConGradU

The iteration for **ConGradU** results in

$$x^{j+1} = \operatorname{argmax}\{x^{jT}Ax : \|x\|_2 = 1,\ \|x\|_0 \leq k\} = \frac{T_k(Ax^j)}{\|T_k(Ax^j)\|_2},\ j = 0, \ldots$$

- **Convergence:** Since the objective is continuously differentiable, by previous result, we have here that every limit point of the sequence $\{x^j\}$ converges to a stationary point.
- **Complexity:** $O(kn)$ or $O(mn)$.

l_0-constrained PCA via ConGradU

The iteration for **ConGradU** results in

$$x^{j+1} = \operatorname{argmax}\{x^{jT}Ax : \|x\|_2 = 1,\ \|x\|_0 \le k\} = \frac{T_k(Ax^j)}{\|T_k(Ax^j)\|_2},\ j = 0, \ldots$$

- **Convergence:** Since the objective is continuously differentiable, by previous result, we have here that every limit point of the sequence $\{x^j\}$ converges to a stationary point.
- **Complexity:** $O(kn)$ or $O(mn)$.
- **The original l_0-constrained problem** can be solved using **ConGradU** with the same complexity as when applied to solving modified problems!
- **Penalized/modified problems require tuning** a tradeoff penalty parameter to get the desired sparsity. This can be computationally very expensive, and is not needed in our scheme.

Back to Q1 – ....All via ConGradU

- All currently known cheap schemes are particular realization of ConGradU
- Novel Schemes can be derived via ConGradU

All we need is a simple toolbox...

Answer to Q1: A Simple ToolBox

All previously listed algorithms are particular realizations of ConGradU.

- **Proposition 1** Given $a \in \mathbf{R}^n, s > 0$,

$$\max_{\|x\|_2=1} \{\langle a, x\rangle^2 - s\|x\|_0\} = \sum_{i=1}^{n} (a_i^2 - s)_+, \; x_i^* = \frac{a_i[\operatorname{sgn}(a_i^2 - s)]_+}{\sqrt{\sum_{j=1}^n a_j^2[\operatorname{sgn}(a_j^2 - s)]_+}}.$$

Answer to Q1: A Simple ToolBox

All previously listed algorithms are particular realizations of ConGradU.

- **Proposition 1** Given $a \in \mathbf{R}^n, s > 0$,

$$\max_{\|x\|_2=1} \{\langle a, x\rangle^2 - s\|x\|_0\} = \sum_{i=1}^{n} (a_i^2 - s)_+, \; x_i^* = \frac{a_i[\operatorname{sgn}(a_i^2 - s)]_+}{\sqrt{\sum_{j=1}^n a_j^2[\operatorname{sgn}(a_j^2 - s)]_+}}.$$

- **Proposition 2** For $a \in \mathbf{R}^n$, $w \in \mathbf{R}^n_{++}$, and $W = \operatorname{diag}(w)$

$$\max_{\|x\|_2 \leq 1} \{\langle a, x\rangle - \|Wx\|_1\} = \|S_w(a)\|, \; x^* = S_w(a)/\|S_w(a)\|_2.$$

$S_w(a) = (|a| - w)_+ \operatorname{sgn}(a)$. (Soft Threshold)

Answer to Q1: A Simple ToolBox

All previously listed algorithms are particular realizations of ConGradU.

- **Proposition 1** Given $a \in \mathbf{R}^n, s > 0$,

$$\max_{\|x\|_2=1} \{\langle a, x\rangle^2 - s\|x\|_0\} = \sum_{i=1}^{n} (a_i^2 - s)_+,\ x_i^* = \frac{a_i[\mathrm{sgn}(a_i^2 - s)]_+}{\sqrt{\sum_{j=1}^n a_j^2[\mathrm{sgn}(a_j^2 - s)]_+}}.$$

- **Proposition 2** For $a \in \mathbf{R}^n$, $w \in \mathbf{R}^n_{++}$, and $W = \mathrm{diag}(w)$

$$\max_{\|x\|_2\le 1} \{\langle a, x\rangle - \|Wx\|_1\} = \|S_w(a)\|,\ x^* = S_w(a)/\|S_w(a)\|_2.$$

$S_w(a) = (|a| - w)_+\mathrm{sgn}(a)$. (Soft Threshold)

- **Proposition 3** Given $a \in \mathbf{R}^n$, we have

$$\max\{\langle a, x\rangle : \|x\|_2 \le 1, \|x\|_1 \le k, x \in \mathbf{R}^n\} = \min\{\lambda k + \|S_{\lambda e}(a)\|_2 : \lambda \in \mathbb{R}_+\}$$

Moreover, if λ solves the one-dimensional dual, then an optimal solution

$$x^*(\lambda) = S_{\lambda e}(a)/\|S_{\lambda e}(a)\|_2,\ (e \equiv (1, \ldots, 1) \in \mathbf{R}^n).$$

Nonsmooth Convex Reformulations

D'aspremont et al. (08), Journee et al. (10)

l_0**-penalized PCA problem:** $\max\{x^T Ax - s\|x\|_0 : \|x\|_2 \leq 1, x \in \mathbf{R}^n\}$

Exploiting A PSD $A := B^T B$ with $B \in \mathbf{R}^{m\times n}$, yields

$$\max\{\|Bx\|_2^2 - s\|x\|_0 : \|x\|_2 \leq 1, x \in \mathbf{R}^n\}.$$

Nonsmooth Convex Reformulations

D'aspremont et al. (08), Journee et al. (10)

l_0**-penalized PCA problem:** $\max\{x^T A x - s\|x\|_0 : \|x\|_2 \leq 1, x \in \mathbf{R}^n\}$

Exploiting A PSD $A := B^T B$ with $B \in \mathbf{R}^{m \times n}$, yields

$$\max\{\|Bx\|_2^2 - s\|x\|_0 : \|x\|_2 \leq 1, x \in \mathbf{R}^n\}.$$

The objective is neither concave nor convex. Using the simple fact $\|Bx\|_2^2 = \max_{\|z\|_2 \leq 1}\{\langle z, Bx\rangle^2\}$, the problem is equivalent to

$$\max_{\|x\|_2 \leq 1} \max_{\|z\|_2 \leq 1} \left\{\langle z, Bx\rangle^2 \ - s\|x\|_0\right\} = \max_{\|z\|_2 \leq 1} \max_{\|x\|_2 \leq 1} \left\{\langle B^T z, x\rangle^2 \ - s\|x\|_0\right\}.$$

Nonsmooth Convex Reformulations

D'aspremont et al. (08), Journee et al. (10)

l_0-**penalized PCA problem:** $\max\{x^T Ax - s\|x\|_0 : \|x\|_2 \leq 1, x \in \mathbf{R}^n\}$

Exploiting A PSD $A := B^T B$ with $B \in \mathbf{R}^{m\times n}$, yields

$$\max\{\|Bx\|_2^2 - s\|x\|_0 : \|x\|_2 \leq 1, x \in \mathbf{R}^n\}.$$

The objective is neither concave nor convex. Using the simple fact $\|Bx\|_2^2 = \max_{\|z\|_2\leq 1}\{\langle z, Bx\rangle^2\}$, the problem is equivalent to

$$\max_{\|x\|_2\leq 1}\max_{\|z\|_2\leq 1}\{\langle z, Bx\rangle^2 - s\|x\|_0\} = \max_{\|z\|_2\leq 1}\max_{\|x\|_2\leq 1}\{\langle B^T z, x\rangle^2 - s\|x\|_0\}.$$

Now, the inner minimization in x can be solved (use **P1**):

$$\max_{x\in\mathbf{R}^n}\{\|Bx\|_2^2 - s\|x\|_0 : \|x\|_2 \leq 1\} = \max_{z\in\mathbf{R}^m}\{\sum_{i=1}^n [\langle b_i, z\rangle^2 - s]_+ : \|z\|_2 \leq 1\}$$

where $b_i \in \mathbf{R}^m$ is the i^{th} column of B.
Since the objective function $f(z) := \sum_i [\langle b_i, z\rangle^2 - s]_+$ is now clearly convex, we can apply ConGradU, recovering the alg. of Journee et al. (10).

More Examples on NSO Reformulation

Similarly, for the l_1-penalized PCA problem one can show:

$$\max\{x^T Ax - s\|x\|_1 : \|x\|_2 = 1, x \in \mathbf{R}^n\} = \max_{z \in \mathbf{R}^m}\{\sum_{i=1}^{n}(|b_i^T z| - s)_+^2 : \|z\|_2 \leq 1\}$$

We can now apply ConGradU to the convex objective $f(z) = \sum_i [|b_i^T z| - s]_+^2$, and for which our convergence results for the nonsmooth case hold true.

This recovers exactly the other algorithm of Journee et al. (2010).

More Examples on NSO Reformulation

Similarly, for the l_1-penalized PCA problem one can show:

$$\max\{x^T Ax - s\|x\|_1 : \|x\|_2 = 1, x \in \mathbf{R}^n\} = \max_{z \in \mathbf{R}^m}\{\sum_{i=1}^{n}(|b_i^T z| - s)_+^2 : \|z\|_2 \leq 1\}$$

We can now apply ConGradU to the convex objective $f(z) = \sum_i [|b_i^T z| - s]_+^2$, and for which our convergence results for the nonsmooth case hold true.

This recovers exactly the other algorithm of Journee et al. (2010).

ConGradU is Very Flexible
Tackling more general problems......

A General Class of Problems

$$(G) \qquad \max_x \{f(x) + g(|x|) : x \in C\}$$

$$\begin{array}{ll} f : \mathbf{R}^n \to \mathbf{R} & \text{is convex,} \\ g : \mathbf{R}^n_+ \to \mathbf{R} & \text{is convex differentiable and montonote decreasing} \\ C \subseteq \mathbf{R}^n & \text{is a compact set.} \end{array}$$

Here $|x| := (|x_1|, \ldots, |x_n|)^T$; monotone decreasing means componentwise.

- Useful for handling penalized/approximate problems.
- Note: the composition $g(|x|)$ is not necessarily convex ...But after a simple transformation we can show that **CondGradU** can be applied to (G), and produces the following simple scheme.

A Simple Scheme for Solving (G)

$$(G) \qquad \max_x \{f(x) + g(|x|) : x \in C\}$$

A-weighted l_1-norm maximization problem:

$$x^0 \in C,\ x^{j+1} = \operatorname{argmax}\{\langle a^j, x\rangle - \sum_i w_i^j |x_i| : x \in C\},\ j = 0, \ldots,$$

where $w^j := -g'(|x^j|) > 0$ and $a^j := f'(x^j) \in \mathbf{R}^n$.

A Simple Scheme for Solving (G)

$$(G) \qquad \max_x \{f(x) + g(|x|) : x \in C\}$$

A-weighted l_1-norm maximization problem:

$$x^0 \in C,\ x^{j+1} = \operatorname{argmax}\{\langle a^j, x\rangle - \sum_i w_i^j |x_i| : x \in C\},\ j = 0, \ldots,$$

where $w^j := -g'(|x^j|) > 0$ and $a^j := f'(x^j) \in \mathbf{R}^n$.

For *penalized/approximate penalized SPCA*, C is a unit ball, and above admits a **closed form solution** thanks to **P2** seen before:

$$x^{j+1} = \frac{S_{w^j}(f'(x^j))}{\|S_{w^j}(f'(x^j))\|},\ j = 0, \ldots$$

Example I – A Novel Direct Approach for l_1-penalized SPCA via (G)

$$\max\{x^T Ax - s\|x\|_1 : \|x\|_2 = 1, x \in \mathbf{R}^n\}, (s > 0)$$

Using our results, applying ConGradU reduces to

$$x^{j+1} = \frac{S_{se}(A_\sigma x^j)}{\|S_{se}(A_\sigma x^j)\|_2},\ e \equiv (1, \ldots, 1)$$

and $S_w(a) = \underset{x}{\operatorname{argmin}}\{\frac{1}{2}\|x - a\|_2^2 + \|Wx\|_1\} = (|a| - w)_+ \operatorname{sgn}(a)$.

- This approach can handle matrices A that are not positive semidefinite (by taking $\sigma > 0, A_\sigma := A + \sigma I_n$).
- In fact, **any *other* convex $f(\cdot)$ can be used!**
- Allows for stronger convergence results than when applying the conditional gradient method to the nonsmooth equivalent reformulation.

Example II : The Approximate l_0-penalized PCA Problem

$$\max\{x^T Ax - s\|x\|_0 : \|x\|_2 = 1, x \in \mathbf{R}^n\}, (s > 0).$$

- Approximations of the l_0 norm by some nicer continuous functions have been considered in various contexts, e.g., machine learning [Mangasarian (96), West (03)]; ... Compressed sensing [Borwein-Luke (11)] .
- Naturally emerged from very well-known mathematical approximations of the step and sign functions Bracewell (2000). Formally, we want to replace the problematic expression sgn $(|t|)$ by some nicer function

$$\|x\|_0 = \sum_{i=1}^{n} \operatorname{sgn}(|x_i|) = \lim_{p\to 0} \sum_{i=1}^{n} \varphi_p(|x_i|)$$

where $\varphi_p : \mathbf{R}_+ \to \mathbf{R}_+$ is an appropriately chosen smooth concave functions, monotone increasing and normalized such that $\varphi_p(0) = 0, \varphi_p'(0) > 0$.

Example II : The Approximate l_0-penalized PCA Problem

$$\max\{x^T Ax - s\|x\|_0 : \|x\|_2 = 1, x \in \mathbf{R}^n\}, (s > 0).$$

- Approximations of the l_0 norm by some nicer continuous functions have been considered in various contexts, e.g., machine learning [Mangasarian (96), West (03)]; ... Compressed sensing [Borwein-Luke (11)] .
- Naturally emerged from very well-known mathematical approximations of the step and sign functions Bracewell (2000). Formally, we want to replace the problematic expression sgn $(|t|)$ by some nicer function

$$\|x\|_0 = \sum_{i=1}^{n} \operatorname{sgn}(|x_i|) = \lim_{p \to 0} \sum_{i=1}^{n} \varphi_p(|x_i|)$$

where $\varphi_p : \mathbf{R}_+ \to \mathbf{R}_+$ is an appropriately chosen smooth concave functions, monotone increasing and normalized such that $\varphi_p(0) = 0, \varphi_p'(0) > 0$.

- The resulting *approximate* l_0-penalized PCA is in the form (G):

$$\max\{x^T Ax - s\sum_{i=1}^{n} \varphi_p(|x_i|) : \|x\|_2 = 1, x \in \mathbf{R}^n\},\ (s > 0, p > 0).$$

Examples of Concave $\varphi_p(\cdot), p > 0$ Approximations for $\|x\|_0$

1. $\varphi_p(t) = (2/\pi)\tan^{-1}(t/p)$,
2. $\varphi_p(t) = \log(1 + t/p)/\log(1 + 1/p)$,
3. $\varphi_p(t) = (1 + p/t)^{-1}$,
4. $\varphi_p(t) = 1 - e^{-t/p}$. A nice feature:it also lower bounds l_0,

$\sum_{i=1}^{n} \varphi_p(|x_i|) \leq \|x\|_0, \quad \forall x \in \mathbf{R}^n$.

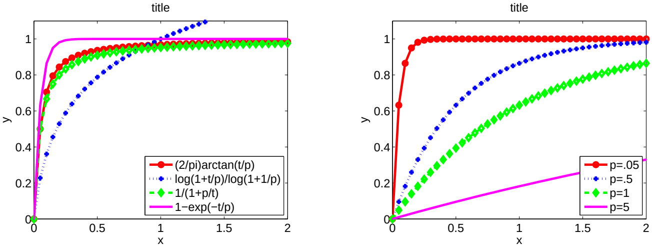


Figure : The left plot $\varphi_p(t)$ for fixed $p = .05$. The right plot how concave approximation $1 - e^{-t/p}$ converges to the indicator function as $p \to 0$.

Some Simulations – Random Matrices -[For more see the paper]

- Our goal is to solve very large sparse PCA problems. The largest dimension we approach is $n = 50000$.
- However, the ConGradU algorithm applied to l_0-constrained PCA has a very cheap $O(mn)$ iterations and is limited only by storage of a data matrix.
- Thus, on larger computers, extremely large-scale sparse PCA problems (much larger than those solved even here) are also feasible.

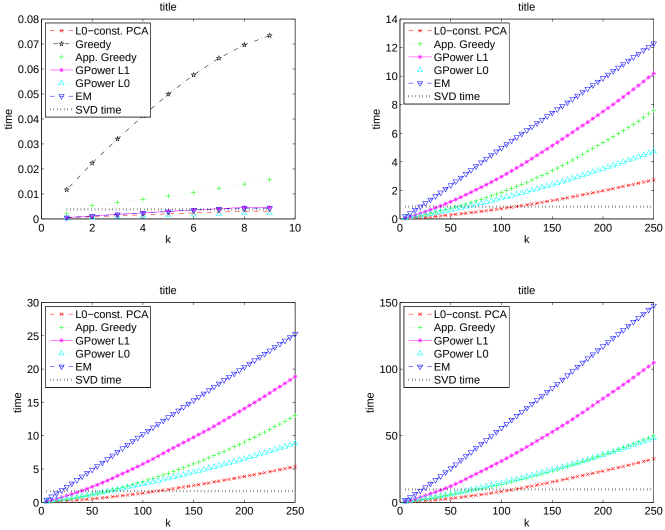
Some Simulations – Random Matrices -[For more see the paper]

- Our goal is to solve very large sparse PCA problems. The largest dimension we approach is $n = 50000$.
- However, the ConGradU algorithm applied to l_0-constrained PCA has a very cheap $O(mn)$ iterations and is limited only by storage of a data matrix.
- Thus, on larger computers, extremely large-scale sparse PCA problems (much larger than those solved even here) are also feasible.

- We here consider random data matrices $F \in \mathbf{R}^{m \times n}$ with $F_{ij} \sim N(0, 1/m)$.
- The experiments consider $n = 10$ ($m = 6$) and $n = 5000, 10000, 50000$ (each with $m = 150$), each using 100 simulations.
- We consider l_0-constrained PCA with $k = 2, \ldots, 9$ for $n = 10$ and $k = 5, 10, \ldots, 250$ for the remaining tests.
- The svdTime is the time required to compute the principal eigenvector of $F^T F$ which is used to compute an initial solution for l_0-constrained PCA.
- Comparison of **ConGradU**: with l_0, l_1 penalized version(GPower of Journee et al.) and EM for l_1-constrained.

Average Time to Produce Sparse Eigenvectors of $F^T F$

$$A = F^T F \text{ with } F \in \mathbf{R}^{m \times n} \text{ with } F_{ij} \sim N(0, 1/m)$$



Summary and Extensions

Problem structures beneficially exploited to build one very simple scheme **ConGradU**:

- Encompasses all currently known cheap methods for sparse PCA..and more..
- Can be applied just as easily to solve the **original l_0-constrained problem**
- All of the cheap algorithms give similar performance. When desired sparsity is known, our novel scheme appears as the cheapest
- **Caveat:** None of currently known algorithms provide certificate/bounds to global optimality for the original SPCA.

Summary and Extensions

Problem structures beneficially exploited to build one very simple scheme **ConGradU**:

- Encompasses all currently known cheap methods for sparse PCA..and more..
- Can be applied just as easily to solve the **original l_0-constrained problem**
- All of the cheap algorithms give similar performance. When desired sparsity is known, our novel scheme appears as the cheapest
- **Caveat:** None of currently known algorithms provide certificate/bounds to global optimality for the original SPCA.

Our tools can be easily used to produce novel simple algorithms for tackling directly other similar problems, (details in our paper). For example:

1. Sparse Singular Value Decomposition:

$$\max \{x^T By : \|x\|_2 = 1, \ \|y\|_2 = 1, \ \|x\|_0 \le k_1, \ \|y\|_0 \le k_2\}$$

2. Sparse Canonical Correlation Analysis:

$$\max \{x^T B^T Cy : x^T B^T Bx = 1 \ y^T C^T Cy = 1, \ \|x\|_0 \le k_1, \ \|y\|_0 \le k_2\}$$

3. Sparse PCA with other convex objectives $f(\cdot)$ or/and additonal "simple" constraints:

$$\max \{f(x) : \|x\|_2 = 1, \ \|x\|_0 \le k, x \in \mathcal{C}\}$$

For More Details, Results....

R. Luss and M. Teboulle. Conditional Gradient Algorithms for Rank-One Matrix Approximations with a Sparsity Constraint.

SIAM Review, (2013). In Press

For More Details, Results....

R. Luss and M. Teboulle. Conditional Gradient Algorithms for Rank-One Matrix Approximations with a Sparsity Constraint.

SIAM Review, (2013). In Press

Thank you for listening!