Sparse Principal Component Analysis via Alternating Maximization and Efficient Parallel Implementations

Martin Takáč

The University of Edinburgh

Joint work with

- Peter Richtárik (Edinburgh University)
- Selin Damla Ahipasaoglu (Singapore University of Technology and Design)

Based on:

Alternating Maximization: Unifying Framework for 8 Sparse PCA Formulations and Efficient Parallel Codes (http://arxiv.org/abs/1212.4137)

Optimization and Big Data, May 1–3, 2013

Overview

- Where is PCA useful?
- Why Sparse PCA?
- Different formulations for SPCA
- Alternating maximization algorithm
- Parallel implementations
- 24AM library
- Numerical experiments

What is Principal Component Analysis (PCA)?

PCA is a tool used for factor analysis and dimension reduction in virtually all areas of science and engineering, e.g.:

- machine learning
- statistics
- genetics
- finance
- computer networks

What is Principal Component Analysis (PCA)?

PCA is a tool used for factor analysis and dimension reduction in virtually all areas of science and engineering, e.g.:

- machine learning
- statistics
- genetics
- finance
- computer networks

Let $A \in \mathbb{R}^{n \times p}$ denote a data matrix encoding n samples (observations) of p variables (features).

PCA aims to extract a few linear combinations of the columns of A, called principal components (PCs), pointing in mutually orthogonal directions, together explaining as much variance in the data as possible.

Finding Leading Principal Components (PC)

The first PC is obtained by solving

$$\max\{\mathbf{Var}\{x^T A\} : \|x\|_2 = 1\} = \max\{\|Ax\|^2 : \|x\|_2 = 1\}, \qquad (1)$$

where $\|\cdot\|$ is a suitable norm for measuring variance

- classical PCA employs the L_2 norm in the objective
- robust PCA uses the L_1 norm

Finding Leading Principal Components (PC)

The first PC is obtained by solving

$$\max\{\mathbf{Var}\{x^T A\} : \|x\|_2 = 1\} = \max\{\|Ax\|^2 : \|x\|_2 \leq 1\}, \qquad (1)$$

where $\|\cdot\|$ is a suitable norm for measuring variance

- classical PCA employs the L_2 norm in the objective
- robust PCA uses the L_1 norm

Some terminology:

- the solution x of (1) is called the **loading vector**
- Ax (normalized) is the first PC

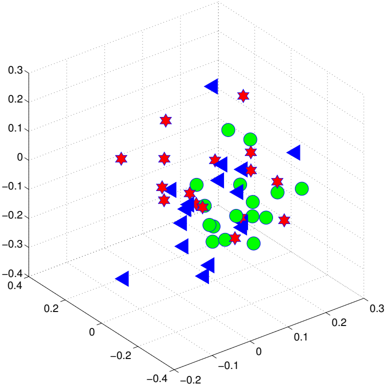
Further PCs can be obtained in the same way with A replaced by a new matrix in a process called **deflation**. For example the second PC can be found by solving (1) with a new matrix $A := A(1 - x_1 x_1^T)$, where x_1 is the first loading vector.

Using PCA for Visualisation

We have 16 images in each of 3 different categories.
Each image is "somehow" represented by a vector $x \in \mathbb{R}^{5,000}$.
Our Task: We would like to visualize these images in 3D space

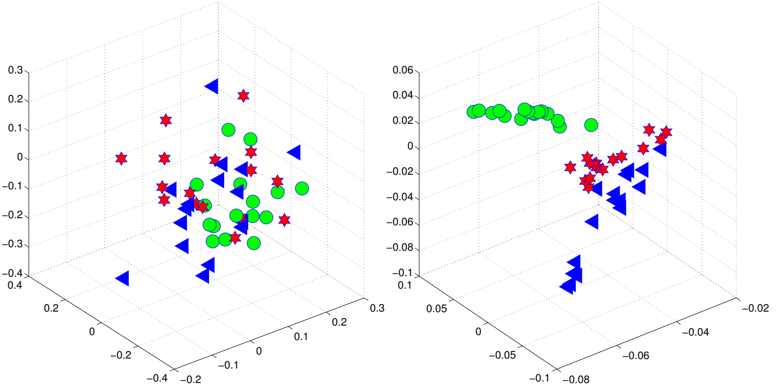
Using PCA for Visualisation

We have 16 images in each of 3 different categories.
Each image is "somehow" represented by a vector $x \in \mathbb{R}^{5,000}$.
Our Task: We would like to visualize these images in 3D space



Using PCA for Visualisation

We have 16 images in each of 3 different categories.
Each image is "somehow" represented by a vector $x \in \mathbb{R}^{5,000}$.
Our Task: We would like to visualize these images in 3D space



Random projection to 3D (left) vs. projection onto 3 loading vectors obtained by PCA (right)

Why Sparse PCA?

- loading vectors obtained by PCA are almost always **dense**
- sometimes sparse loading vectors are desirable to enhance the **interpretability** of the components and are **easier to store**

Why Sparse PCA?

- loading vectors obtained by PCA are almost always **dense**
- sometimes sparse loading vectors are desirable to enhance the **interpretability** of the components and are **easier to store**

Example:
Assume we have n newspaper articles with a total of p distinct words. We can build a matrix $A \in \mathbb{R}^{n \times p}$ such that $A_{j,i}$ **counts the number of appearances of word i in article j.**
After some scaling and normalization we can apply SPCA. Now, **non-zero values in the loading vector can be associated with words** – those words can be used to characterize articles – for the result you have to wait for a few slides :)

Zhang, Y., El Ghaoui, L. *Large-scale sparse principal component analysis with application to text data*, Advances in Neural Information Processing Systems **24**:532-539, 2011

How can we Enforce Sparsity?

How can we Enforce Sparsity?

Adding a Penalty to an Objective Function

Let $\mathcal{P}(x)$ be a sparsity inducing penalty

$$\max\{\|Ax\| - \gamma\mathcal{P}(x) : \|x\|_2 \leq 1\}, \quad \gamma > 0$$

How can we Enforce Sparsity?

Adding a Penalty to an Objective Function

Let $\mathcal{P}(x)$ be a sparsity inducing penalty

$$\max\{\|Ax\| - \gamma\mathcal{P}(x) : \|x\|_2 \leq 1\}, \quad \gamma > 0$$

Adding a Sparsity Inducing Constraint

Let $\mathcal{C}(x)$ be a sparsity inducing constraint

$$\max\{\|Ax\| : \|x\|_2 \leq 1, \mathcal{C}(x) \leq k\}, \quad k > 0$$

How can we Enforce Sparsity?

Adding a Penalty to an Objective Function

Let $\mathcal{P}(x)$ be a sparsity inducing penalty

$$\max\{\|Ax\| - \gamma\mathcal{P}(x) : \|x\|_2 \leq 1\}, \quad \gamma > 0$$

Adding a Sparsity Inducing Constraint

Let $\mathcal{C}(x)$ be a sparsity inducing constraint

$$\max\{\|Ax\| : \|x\|_2 \leq 1, \mathcal{C}(x) \leq k\}, \quad k > 0$$

Candidates for $\mathcal{P}(x)$ and $\mathcal{C}(x)$:

- $\|x\|_1 = \sum_{i=1}^{p} |x_i|$
- $\|x\|_0 = |\{i : x_i \neq 0\}|$

Eight Sparse PCA Optimization Formulations

$$OPT = \max_{x \in X} f(x), \tag{2}$$

#	Var.	SI	SI usage	X	$f(x)$
1	L_2	L_0	const.	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \le 1, \ \|x\|_0 \le s\}$	$\|Ax\|_2$
2	L_1	L_0	const.	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \le 1, \ \|x\|_0 \le s\}$	$\|Ax\|_1$
3	L_2	L_1	const.	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \le 1, \ \|x\|_1 \le \sqrt{s}\}$	$\|Ax\|_2$
4	L_1	L_1	const.	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \le 1, \ \|x\|_1 \le \sqrt{s}\}$	$\|Ax\|_1$
5	L_2	L_0	pen.	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \le 1\}$	$\|Ax\|_2^2 - \gamma\|x\|_0$
6	L_1	L_0	pen.	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \le 1\}$	$\|Ax\|_1^2 - \gamma\|x\|_0$
7	L_2	L_1	pen.	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \le 1\}$	$\|Ax\|_2 - \gamma\|x\|_1$
8	L_1	L_1	pen.	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \le 1\}$	$\|Ax\|_1 - \gamma\|x\|_1$

Note: All our optimization problems are **NOT** convex problems!

How do we Solve the SPCA Problem?

Alternating Maximization Algorithm (AM)

Suppose we have the following optimization problem

$$\max_{x \in X} \max_{y \in Y} F(x, y) \tag{3}$$

How do we Solve the SPCA Problem?

Alternating Maximization Algorithm (AM)

Suppose we have the following optimization problem

$$\max_{x \in X} \max_{y \in Y} F(x, y) \tag{3}$$

Alternating Maximization Algorithm

Select initial point $x^{(0)} \in \mathbb{R}^p$; $k \leftarrow 0$
Repeat
  $y^{(k)} \leftarrow y(x^{(k)}) := \arg\max_{y \in Y} F(x^{(k)}, y)$
  $x^{(k+1)} \leftarrow x(y^{(k)}) := \arg\max_{x \in X} F(x, y^{(k)})$
Until a stopping criterion is satisfied

How do we Solve the SPCA Problem?

Alternating Maximization Algorithm (AM)

Suppose we have the following optimization problem

$$\max_{x \in X} \max_{y \in Y} F(x, y) \tag{3}$$

Alternating Maximization Algorithm

Select initial point $x^{(0)} \in \mathbb{R}^p$; $k \leftarrow 0$
Repeat
$y^{(k)} \leftarrow y(x^{(k)}) := \arg\max_{y \in Y} F(x^{(k)}, y)$
$x^{(k+1)} \leftarrow x(y^{(k)}) := \arg\max_{x \in X} F(x, y^{(k)})$
Until a stopping criterion is satisfied

All we have to do now is to show that (2) *can be reformulated as* (3) *and then apply AM algorithm!*

Problem Reformulations

#	X	Y	$F(x,y)$
1	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \leq 1, \ \|x\|_0 \leq s\}$	$\{y \in \mathbb{R}^n \ : \ \|y\|_2 \leq 1\}$	$y^T Ax$
2	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \leq 1, \ \|x\|_0 \leq s\}$	$\{y \in \mathbb{R}^n \ : \ \|y\|_\infty \leq 1\}$	$y^T Ax$
3	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \leq 1, \ \|x\|_1 \leq \sqrt{s}\}$	$\{y \in \mathbb{R}^n \ : \ \|y\|_2 \leq 1\}$	$y^T Ax$
4	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \leq 1, \ \|x\|_1 \leq \sqrt{s}\}$	$\{y \in \mathbb{R}^n \ : \ \|y\|_\infty \leq 1\}$	$y^T Ax$
5	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \leq 1\}$	$\{y \in \mathbb{R}^n \ : \ \|y\|_2 \leq 1\}$	$(y^T Ax)^2 - \gamma\|x\|_0$
6	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \leq 1\}$	$\{y \in \mathbb{R}^n \ : \ \|y\|_\infty \leq 1\}$	$(y^T Ax)^2 - \gamma\|x\|_0$
7	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \leq 1\}$	$\{y \in \mathbb{R}^n \ : \ \|y\|_2 \leq 1\}$	$y^T Ax - \gamma\|x\|_1$
8	$\{x \in \mathbb{R}^p \ : \ \|x\|_2 \leq 1\}$	$\{y \in \mathbb{R}^n \ : \ \|y\|_\infty \leq 1\}$	$y^T Ax - \gamma\|x\|_1$

Example #1: L0 constrained L2 PCA

$$\max_{\|x\|_2 \leq 1, \|x\|_0 \leq s} \max_{\|y\|_2 \leq 1} y^T Ax = \max_{\|x\|_2 \leq 1, \|x\|_0 \leq s} \frac{1}{\|Ax\|_2}(Ax)^T Ax = \max_{\|x\|_2 \leq 1, \|x\|_0 \leq s} \|Ax\|_2$$

Example #2: L0 constrained L1 PCA

$$\max_{\|x\|_2 \leq 1, \|x\|_0 \leq s} \max_{\|y\|_\infty \leq 1} y^T Ax = \max_{\|x\|_2 \leq 1, \|x\|_0 \leq s} \sum_{j=1}^{n} |A_j x| = \max_{\|x\|_2 \leq 1, \|x\|_0 \leq s} \|Ax\|_1,$$

where A_j is the j-th row of the matrix A

AM Algorithm for SPCA

Select initial point $x^{(0)} \in \mathbb{R}^p$; $k \leftarrow 0$
Repeat
  $u = Ax^{(k)}$
    If L_1 variance **then** $y^{(k)} \leftarrow \operatorname{sgn}(u)$
    If L_2 variance **then** $y^{(k)} \leftarrow u/\|u\|_2$
  $v = A^T y^{(k)}$
    If L_0 penalty **then** $x^{(k+1)} \leftarrow U_\gamma(v)/\|U_\gamma(v)\|_2$
    If L_1 penalty **then** $x^{(k+1)} \leftarrow V_\gamma(v)/\|V_\gamma(v)\|_2$
    If L_0 constraint **then** $x^{(k+1)} \leftarrow T_s(v)/\|T_s(v)\|_2$
    If L_1 constraint **then** $x^{(k+1)} \leftarrow V_{\lambda_s(v)}(v)/\|V_{\lambda_s(v)}(v)\|_2$
  $k \leftarrow k+1$
Until a stopping criterion is satisfied

- $(U_\gamma(z))_i := z_i[\operatorname{sgn}(z_i^2 - \gamma)]_+$
- $(V_\gamma(z))_i := \operatorname{sgn}(z_i)(|z_i| - \gamma)_+$
- $T_s(z)$ is hard thresholding operator
- $\lambda_s(z) := \arg\min_{\lambda \geq 0} \lambda\sqrt{s} + \|V_\lambda(z)\|_2$

AM Algorithm for SPCA

Select initial point $x^{(0)} \in \mathbb{R}^p$; $k \leftarrow 0$
Repeat
  $u = Ax^{(k)}$
    If L_1 variance **then** $y^{(k)} \leftarrow \text{sgn}(u)$
    If L_2 variance **then** $y^{(k)} \leftarrow u/\|u\|_2$
  $v = A^T y^{(k)}$
    If L_0 penalty **then** $x^{(k+1)} \leftarrow U_\gamma(v)/\|U_\gamma(v)\|_2$
    If L_1 penalty **then** $x^{(k+1)} \leftarrow V_\gamma(v)/\|V_\gamma(v)\|_2$
    If L_0 constraint **then** $x^{(k+1)} \leftarrow T_s(v)/\|T_s(v)\|_2$
    If L_1 constraint **then** $x^{(k+1)} \leftarrow V_{\lambda_s(v)}(v)/\|V_{\lambda_s(v)}(v)\|_2$
  $k \leftarrow k + 1$
Until a stopping criterion is satisfied

Example #2: L0 constrained L1 PCA

$$\max_{\|x\|_2 \leq 1, \|x\|_0 \leq s} \max_{\|y\|_\infty \leq 1} y^T A x$$

AM Algorithm for SPCA

Select initial point $x^{(0)} \in \mathbb{R}^p$; $k \leftarrow 0$
Repeat
  $u = Ax^{(k)}$
  $y^{(k)} \leftarrow \mathrm{sgn}(u)$
  $v = A^T y^{(k)}$
  $x^{(k+1)} \leftarrow T_s(v)/\|T_s(v)\|_2$
  $k \leftarrow k + 1$
Until a stopping criterion is satisfied

Example #2: L0 constrained L1 PCA

$$\max_{\|x\|_2 \le 1, \|x\|_0 \le s} \max_{\|y\|_\infty \le 1} y^T A x$$

AM Algorithm for SPCA

Select initial point $x^{(0)} \in \mathbb{R}^p$; $k \leftarrow 0$
Repeat
$u = Ax^{(k)}$
$y^{(k)} \leftarrow \mathrm{sgn}(u)$
$v = A^T y^{(k)}$
$x^{(k+1)} \leftarrow T_s(v)/\|T_s(v)\|_2$
$k \leftarrow k+1$
Until a stopping criterion is satisfied

Example #2: L0 constrained L1 PCA - for fixed $\hat{x}$

$$\max_{\|y\|_\infty \leq 1} y^T A\hat{x} \quad \Rightarrow \quad y^* = \mathrm{sgn}(A\hat{x})$$

AM Algorithm for SPCA

Select initial point $x^{(0)} \in \mathbb{R}^p$; $k \leftarrow 0$
Repeat
$u = Ax^{(k)}$
$y^{(k)} \leftarrow \text{sgn}(u)$
$v = A^T y^{(k)}$
$x^{(k+1)} \leftarrow T_s(v)/\|T_s(v)\|_2$
$k \leftarrow k + 1$
Until a stopping criterion is satisfied

Example #2: L0 constrained L1 PCA - for fixed $\hat{y}$

$$\max_{\|x\|_2 \leq 1, \|x\|_0 \leq s} (\hat{y}^T A)x \quad \Rightarrow \quad x^* = T_s(A^T\hat{y})/\|T_s(A^T\hat{y})\|_2$$

Equivalence with GPower Method

- **GPower** (generalized power method) is a simple algorithm for maximizing a convex function Ψ on a compact set Ω, which works via a "linearize and maximize" strategy
- let $\Psi'(z^{(k)})$ be an arbitrary subgradient of Ψ at $z^{(k)}$, then GPower performs the following iteration:

$$z^{(k+1)} = \arg\max_{z\in\Omega}\{\Psi(z^{(k)})+\langle\Psi'(z^{(k)}), z-z^{(k)}\rangle\} = \arg\max_{z\in\Omega}\langle\Psi'(z^{(k)}), z\rangle.$$

Convergence guarantee:

- $\{\Psi(z_k)\}_{k=0}^{\infty}$ is monotonically increasing
- $\Delta_k \leq \frac{\Psi^* - \Psi(z_0)}{k+1}$, where $\Delta_k := \min_{0\leq i\leq k}\{\max_{z\in\Omega}\langle\Psi'(z^{(i)}), z - z^{(i)}\rangle\}$

Journée, M., Nesterov, Y., Richtárik, P. and Sepulchre, R. *Generalized power method for sparse principal component analysis*, Journal of Machine Learning Research, **11**:517-553, 2010

Equivalence with GPower Method

Theorem

The AM and GPower methods are equivalent in the following sense:

1. For the 4 **constrained** sparse PCA formulations, **the** x **iterates** of the AM method applied to the corresponding reformulation are **identical** to the iterates of the GPower method as applied to the problem of maximizing the convex function

   $$F_Y(x) \stackrel{\text{def}}{=} \max_{y \in Y} F(x, y)$$

   on X, started from $x^{(0)}$.

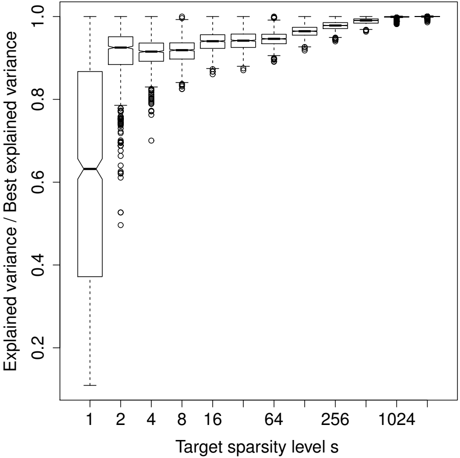
2. For the 4 **penalized** sparse PCA formulations, **the** y **iterates** of the AM method applied to the corresponding reformulation are **identical** to the iterates of the GPower method as applied to the problem of maximizing the convex function

   $$F_X(y) \stackrel{\text{def}}{=} \max_{x \in X} F(x, y)$$

   on Y, started from $y^{(0)}$.

The Hunt for More Explained Variance

- optimization problem (2) is **NOT** convex
- AM finds only a locally optimal solution ⇒ we need more **random** starting points!



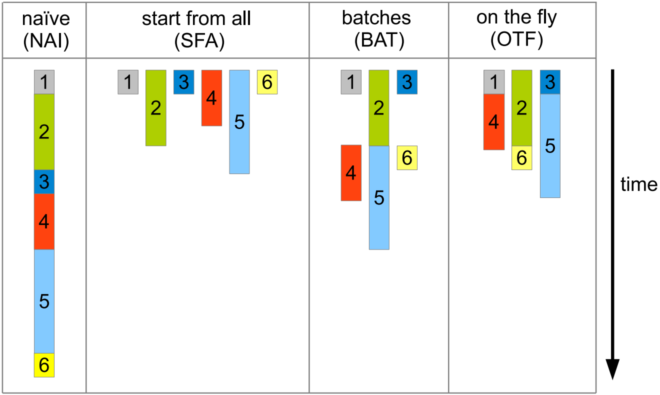
1,000 starting points

Parallel Implementations

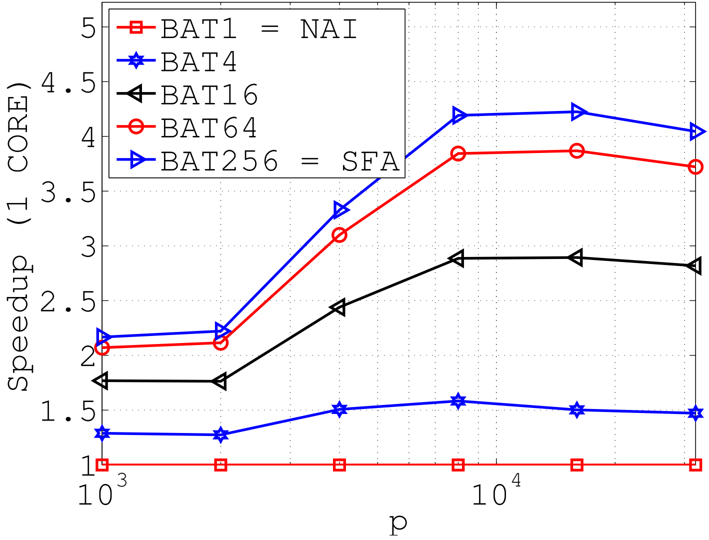
- main computational cost of the algorithm is Matrix-Vector multiplication!
- Matrix-Vector multiplication is BLAS **Level 2 function** and are not implemented in parallel
- we need **more starting points** to improve the **quality** of our "best" local solution

Parallel Implementations

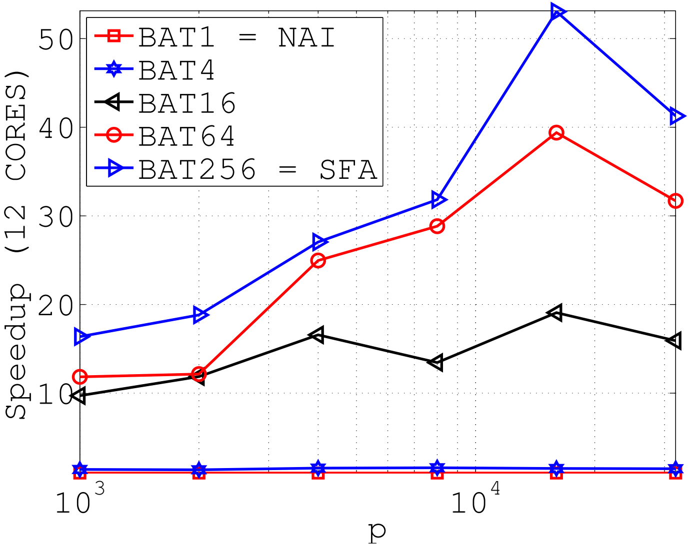
- main computational cost of the algorithm is Matrix-Vector multiplication!
- Matrix-Vector multiplication is BLAS **Level 2 function** and are not implemented in parallel
- we need **more starting points** to improve the **quality** of our “best” local solution



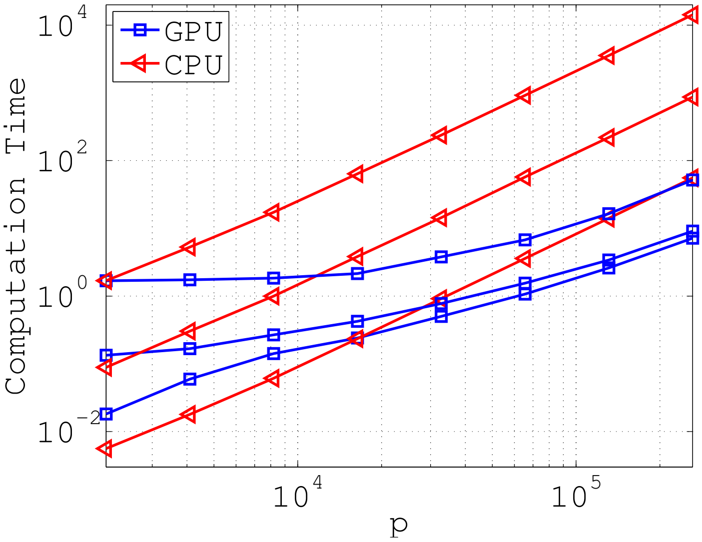
Numerical Experiments - Strategies



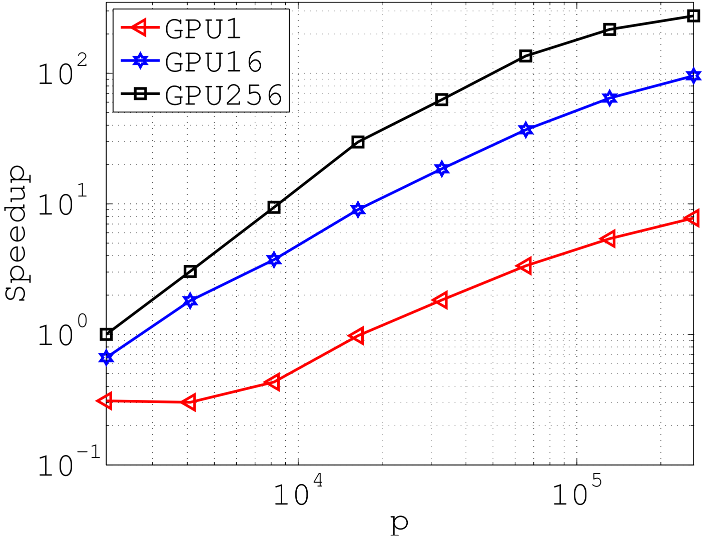
Numerical Experiments - Strategies - 12 cores



Numerical Experiments - GPU



Numerical Experiments - GPU - speedup



Cluster version

$n \times p$	memory	# CPUs	GRID	SP	t_3^1	t_3^4	t_3^{16}
$10^4 \times 2 \cdot 10^5$	14.9 GB	20	10×2	1	0.56	2.06	8.48
$10^4 \times 2 \cdot 10^5$	14.9 GB	20	10×2	32	4.60	18.89	87.84
$10^4 \times 2 \cdot 10^5$	14.9 GB	20	10×2	64	10.47	37.88	166.60
$6 \cdot 10^3 \times 4 \cdot 10^5$	17.8 GB	40	10×4	1	0.78	3.15	9.96
$6 \cdot 10^3 \times 4 \cdot 10^5$	17.8 GB	40	10×4	32	7.39	27.72	125.14
$6 \cdot 10^3 \times 4 \cdot 10^5$	17.8 GB	40	10×4	64	13.19	58.36	201.51
$6 \cdot 10^3 \times 10^6$	44.7 GB	100	10×10	1	0.45	2.44	11.62
$6 \cdot 10^3 \times 10^6$	44.7 GB	100	10×10	32	6.37	29.72	115.73
$6 \cdot 10^3 \times 10^6$	44.7 GB	100	10×10	64	14.14	52.64	219.8
$6 \cdot 10^3 \times 4 \cdot 10^6$	178.8 GB	400	10×40	1	1.24	5.12	31.46
$6 \cdot 10^3 \times 4 \cdot 10^6$	178.8 GB	400	10×40	32	17.50	61.36	255.80
$6 \cdot 10^3 \times 4 \cdot 10^6$	178.8 GB	400	10×40	64	31.36	141.61	525.08
$6 \cdot 10^3 \times 8 \cdot 10^6$	357.6 GB	800	10×80	1	4.14	15.82	95.51
$6 \cdot 10^3 \times 8 \cdot 10^6$	357.6 GB	800	10×80	32	51.11	324.26	619.45
$6 \cdot 10^3 \times 8 \cdot 10^6$	357.6 GB	800	10×80	64	134.89	690.06	-

Numerical Experiments - Large Text Corpora

- we used L_0 constrained L_2 variance formulation (with $s = 5$)
- **Dataset:** news from *New York Times* (102,660 articles, 300,000 words, and approximately 70 million nonzero entries) and abstracts of articles published in *PubMed* (141,043 articles, 8.2 million words, and approximately 484 million nonzeroes)

1st PC	2nd PC	3rd PC	4th PC	5th PC
game	companies	campaign	children	attack
play	company	president	program	government
player	million	al gore	school	official
season	percent	bush	student	US
team	stock	george bush	teacher	united states

1st PC	2nd PC	3rd PC	4th PC	5th PC
disease	cell	activity	cancer	age
level	effect	concentration	malignant	child
patient	expression	control	mice	children
therapy	human	rat	primary	parent
treatment	protein	receptor	tumor	year

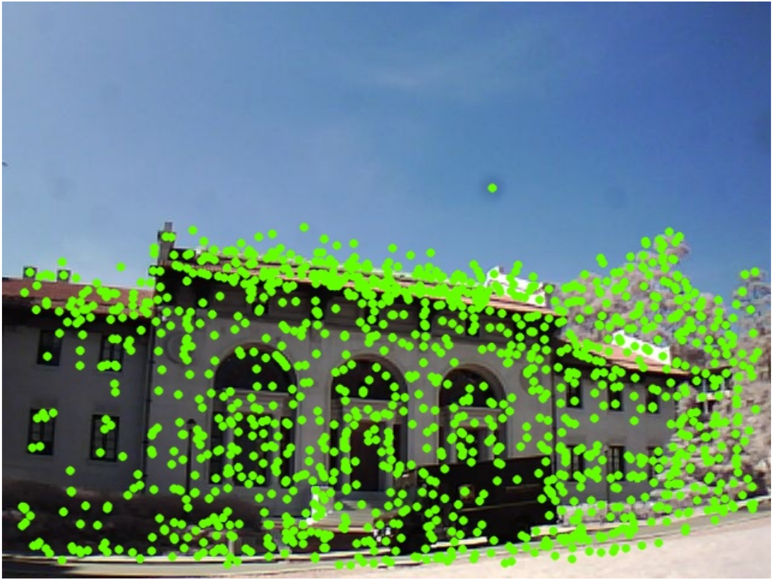
Numerical Experiments - Important Feature Selection

- on each image some features ("words") are identified (by SURF algorithm)
- matrix A is build in the same way as in Large text corpora experiment
- after some scaling and normalization of matrix A we apply SPCA and extract few loading vectors
- we choose only "words" selected by non-zero elements of loading vectors

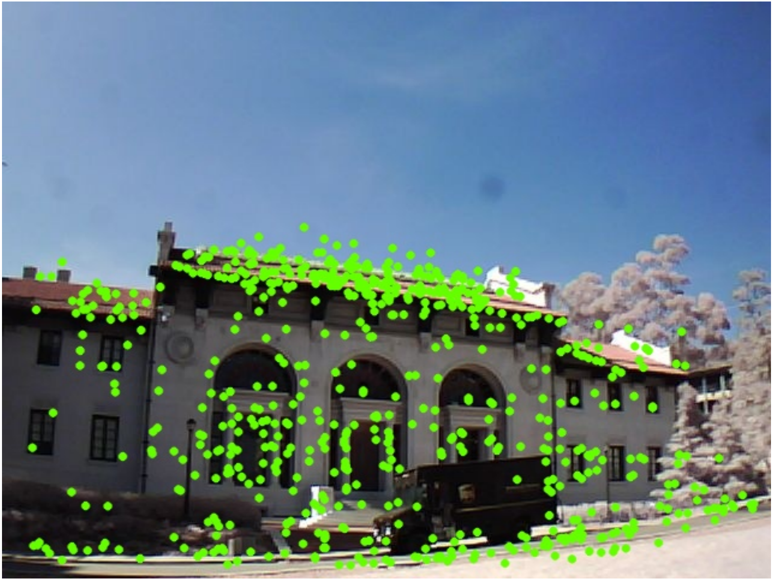
---

Naikal, Nikhil and Yang, Allen Y. and Shankar Sastry, S. *Informative feature selection for object recognition via Sparse PCA*, ICCV '11

Numerical Experiments - Important Feature Selection



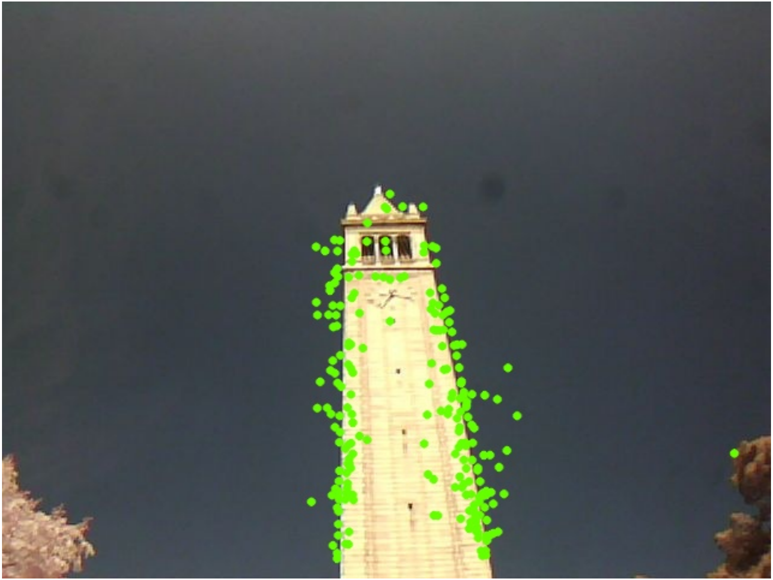
Numerical Experiments - Important Feature Selection



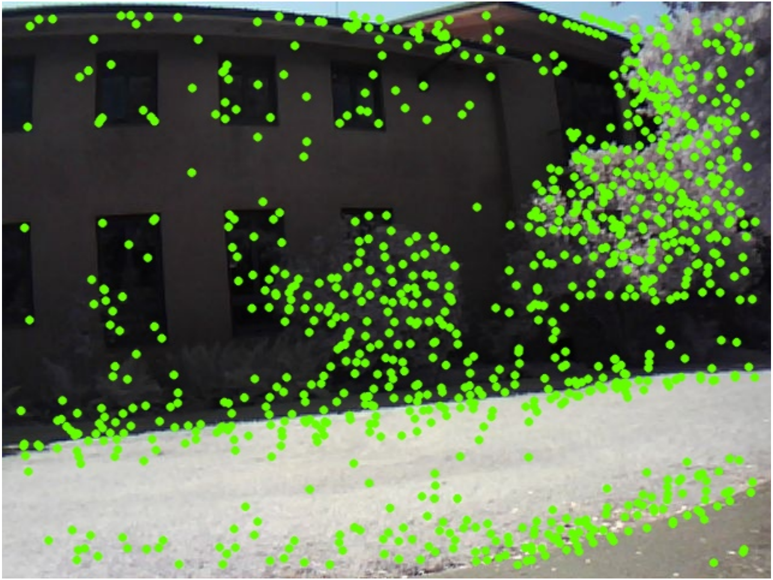
Important Feature Selection - Why does it work?

Numerical Experiments - Important Feature Selection

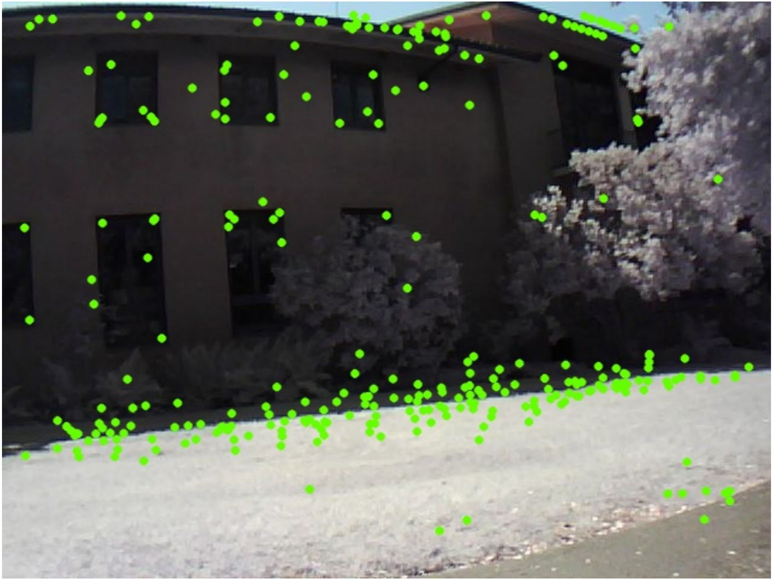
Numerical Experiments - Important Feature Selection



Numerical Experiments - Important Feature Selection



Numerical Experiments - Important Feature Selection



Conclusion

- We applied Alternating Maximization Algorithm for 8 formulations of Sparse PCA
- We implemented all 8 formulations for 3 different architectures (multi-core, GPU and cluster)
- We implements additional strategies (SFA, BAT, NAI, OTF) to facilitate better quality of a solution
- The code is open-source and available at `https://code.google.com/p/24am/`