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:


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
- genetics
- statistics
- 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.
The first PC is obtained by solving

$$\max \{ \text{Var} \{ x^T A \} : \|x\|_2 = 1 \} = \max \{ \|A x\|^2 : \|x\|_2 = 1 \},$$  \hspace{0.5cm} (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
The first PC is obtained by solving

$$\max\{\text{Var}\{x^TA\} : \|x\|_2 = 1\} = \max\{\|Ax\|^2 : \|x\|_2 \leq 1\},$$

(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 $A := A(1 - x_1x_1^T)$, 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

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 :)
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 :)

---

How can we Enforce Sparsity?

Adding a Penalty to an Objective Function

Let $P(x)$ be a sparsity inducing penalty

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

Adding a Sparsity Inducing Constraint

Let $C(x)$ be a sparsity inducing constraint

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

Candidates for $P(x)$ and $C(x)$:

- $\|x\|_1 = \sum_{i=1}^{p} |x_i|$
- $\|x\|_0 = |\{ i : x_i \neq 0 \}|$
How can we Enforce Sparsity?

Adding a Penalty to an Objective Function

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

$$\max \{ \|Ax\| - \gamma \mathcal{P}(x) : \|x\|_2 \leq 1 \}, \quad \gamma > 0$$
How can we Enforce Sparsity?

Adding a Penalty to an Objective Function

Let \( P(x) \) be a sparsity inducing penalty

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

Adding a Sparsity Inducing Constraint

Let \( C(x) \) be a sparsity inducing constraint

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

\begin{align*}
\text{OPT} = \max_{x \in X} f(x),
\end{align*}

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

<table>
<thead>
<tr>
<th>#</th>
<th>Var.</th>
<th>Sl</th>
<th>Sl usage</th>
<th>$X$</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$L_2$</td>
<td>$L_0$</td>
<td>const.</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1, |x|_0 \leq s}$</td>
<td>$|Ax|_2$</td>
</tr>
<tr>
<td>2</td>
<td>$L_1$</td>
<td>$L_0$</td>
<td>const.</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1, |x|_0 \leq s}$</td>
<td>$|Ax|_1$</td>
</tr>
<tr>
<td>3</td>
<td>$L_2$</td>
<td>$L_1$</td>
<td>const.</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1, |x|_1 \leq \sqrt{s}}$</td>
<td>$|Ax|_2$</td>
</tr>
<tr>
<td>4</td>
<td>$L_1$</td>
<td>$L_1$</td>
<td>const.</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1, |x|_1 \leq \sqrt{s}}$</td>
<td>$|Ax|_1$</td>
</tr>
<tr>
<td>5</td>
<td>$L_2$</td>
<td>$L_0$</td>
<td>pen.</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1}$</td>
<td>$|Ax|_2^2 - \gamma |x|_0$</td>
</tr>
<tr>
<td>6</td>
<td>$L_1$</td>
<td>$L_0$</td>
<td>pen.</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1}$</td>
<td>$|Ax|_1^2 - \gamma |x|_0$</td>
</tr>
<tr>
<td>7</td>
<td>$L_2$</td>
<td>$L_1$</td>
<td>pen.</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1}$</td>
<td>$|Ax|_2 - \gamma |x|_1$</td>
</tr>
<tr>
<td>8</td>
<td>$L_1$</td>
<td>$L_1$</td>
<td>pen.</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1}$</td>
<td>$|Ax|_1 - \gamma |x|_1$</td>
</tr>
</tbody>
</table>
How do we Solve the SPCA Problem?

Alternating Maximization Algorithm (AM)
Suppose we have the following optimization problem

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

(2) can be reformulated as (3) and then apply AM algorithm!
How do we Solve the SPCA Problem?

**Alternating Maximization Algorithm (AM)**

Suppose we have the following optimization problem

\[
\max_{x \in X} \max_{y \in Y} F(x, y)
\]  

(3)

**Alternating Maximization Algorithm**

Select initial point \( x^{(0)} \in \mathbb{R}^p; \ k \leftarrow 0 \)

**Repeat**

\[
\begin{align*}
y^{(k)} & \leftarrow y(x^{(k)}) := \arg \max_{y \in Y} F(x^{(k)}, y) \\
x^{(k+1)} & \leftarrow x(y^{(k)}) := \arg \max_{x \in X} F(x, y^{(k)})
\end{align*}
\]

**Until** a stopping criterion is satisfied
How do we Solve the SPCA Problem?

Alternating Maximization Algorithm (AM)
Suppose we have the following optimization problem

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

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

<table>
<thead>
<tr>
<th>#</th>
<th>X</th>
<th>Y</th>
<th>$F(x, y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1, |x|_0 \leq s}$</td>
<td>${y \in \mathbb{R}^n : |y|_2 \leq 1}$</td>
<td>$y^T A x$</td>
</tr>
<tr>
<td>2</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1, |x|_0 \leq s}$</td>
<td>${y \in \mathbb{R}^n : |y|_\infty \leq 1}$</td>
<td>$y^T A x$</td>
</tr>
<tr>
<td>3</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1, |x|_1 \leq \sqrt{s}}$</td>
<td>${y \in \mathbb{R}^n : |y|_2 \leq 1}$</td>
<td>$y^T A x$</td>
</tr>
<tr>
<td>4</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1, |x|_1 \leq \sqrt{s}}$</td>
<td>${y \in \mathbb{R}^n : |y|_\infty \leq 1}$</td>
<td>$y^T A x$</td>
</tr>
<tr>
<td>5</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1}$</td>
<td>${y \in \mathbb{R}^n : |y|_2 \leq 1}$</td>
<td>$(y^T A x)^2 - \gamma |x|_0$</td>
</tr>
<tr>
<td>6</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1}$</td>
<td>${y \in \mathbb{R}^n : |y|_\infty \leq 1}$</td>
<td>$(y^T A x)^2 - \gamma |x|_0$</td>
</tr>
<tr>
<td>7</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1}$</td>
<td>${y \in \mathbb{R}^n : |y|_2 \leq 1}$</td>
<td>$y^T A x - \gamma |x|_1$</td>
</tr>
<tr>
<td>8</td>
<td>${x \in \mathbb{R}^p : |x|_2 \leq 1}$</td>
<td>${y \in \mathbb{R}^n : |y|_\infty \leq 1}$</td>
<td>$y^T A x - \gamma |x|_1$</td>
</tr>
</tbody>
</table>

### Example #1: L0 constrained L2 PCA

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

### Example #2: L0 constrained L1 PCA

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

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 $y^{(k)} \leftarrow \text{sgn}(u)$

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

- $(U_{\gamma}(z))_i := z_i[\text{sgn}(z_i^2 - \gamma)]_+$
- $(V_{\gamma}(z))_i := \text{sgn}(z_i)(|z_i| - \gamma)_+$
- $T_s(z)$ 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: $L_0$ constrained $L_1$ PCA

$$\max \frac{\|x\|_2 \leq 1, \|x\|_0 \leq s}{\|y\|_\infty \leq 1} \max y^T Ax$$
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

\[
\max_{\|x\|_2 \leq 1, \|x\|_0 \leq s} \max_{\|y\|_{\infty} \leq 1} y^T Ax
\]
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 $\hat{x}$

\[
\max_{\|y\|_\infty \leq 1} y^T A\hat{x} \quad \Rightarrow \quad y^* = \text{sgn}(A\hat{x})
\]
AM Algorithm for SPCA

Select initial point $x^{(0)} \in \mathbb{R}^p$; $k \leftarrow 0$

Repeat

\[
u = Ax^{(k)} \\
y^{(k)} \leftarrow \text{sgn}(u) \\
v = A^T y^{(k)} \\
x^{(k+1)} \leftarrow T_s(v)/\|T_s(v)\|_2 \\\nk \leftarrow k + 1
\]

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 $\Psi$ on a compact set $\Omega$, which works via a “linearize and maximize” strategy
- let $\Psi'(z^{(k)})$ be an arbitrary subgradient of $\Psi$ 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 $\Delta_k := \min_{0 \leq i \leq k} \{ \max_{z \in \Omega} \langle \Psi'(z^{(i)}), z - z^{(i)} \rangle \}$

---

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

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

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

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

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 \(\Rightarrow\) we need more random starting points!

![Box plot graph showing explained variance best explained variance for different target sparsity levels and 1,000 starting points.](image)
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

<table>
<thead>
<tr>
<th>naïve (NAI)</th>
<th>start from all (SFA)</th>
<th>batches (BAT)</th>
<th>on the fly (OTF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 2 3 4 5 6</td>
<td>1 2 3</td>
<td>1 4 2 3</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>4 5</td>
<td>4 6</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Numerical Experiments - Strategies

- BAT1 = NAI
- BAT4
- BAT16
- BAT64
- BAT256 = SFA

Graph showing speedup (1 CORE) with different strategies.
Numerical Experiments - Strategies - 12 cores

Speedup (12 CORES)

BAT1 = NAI
BAT4
BAT16
BAT64
BAT256 = SFA

\[
\begin{align*}
10 & \\
3 & \\
10 & \\
4 & \\
10 & \\
20 & \\
30 & \\
40 & \\
50 & \\
p & \\
\end{align*}
\]
Numerical Experiments - GPU

![Graph showing computation time comparison between GPU and CPU]
Numerical Experiments - GPU - speedup

![Graph showing speedup for GPU1, GPU16, and GPU256.

- GPU1
- GPU16
- GPU256

Speedup

$10^0$, $10^1$, $10^2$

$p$

$10^4$, $10^5$]
## Cluster version

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

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

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

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/