A D.C. Programming Approach to the Sparse Generalized Eigenvalue Problem

Bharath K. Sriperumbudur, David A. Torres and Gert R. G. Lanckriet

University of California, San Diego

OPT 2009
Generalized Eigenvalue Problem

Given a matrix pair, \((A, B)\), find a pair \((\lambda, x)\) such that
\[ Ax = \lambda Bx, \]
where \(A, B \in \mathbb{C}^{n \times n}, \mathbb{C}^n \ni x \neq 0\) and \(\lambda \in \mathbb{C}\).

Variational formulation:
\[
\lambda_{\text{max}}(A, B) = \max_{x} \quad x^T A x \\
\text{s.t.} \quad x^T B x = 1,
\]
where \(x \in \mathbb{R}^n, A \in \mathbb{S}^n\) and \(B \in \mathbb{S}_{++}^n\).

- Popular in multivariate statistics and machine learning.
  - **Classification**: Fisher discriminant analysis
  - **Dimensionality reduction**: Principal component analysis, Canonical correlation analysis
  - **Clustering**: Spectral clustering
Applications

- **Fisher Discriminant Analysis (FDA)**
  - $A = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ is the *between-cluster variance*.
  - $B = \Sigma_1 + \Sigma_2$ is the *within-cluster variance*.

- **Principal Component Analysis (PCA)**
  - $A = \Sigma$ is the *covariance matrix*.
  - $B$ is the *identity matrix*.

- **Canonical Correlation Analysis (CCA)**
  - $A = \begin{pmatrix} 0 & S_{xy} \\ S_{yx} & 0 \end{pmatrix}$.
  - $B = \begin{pmatrix} S_{xx} & 0 \\ 0 & S_{yy} \end{pmatrix}$, where $S_{..}$ represents the cross-covariance matrix.
Why Sparsity?

- Usually, the solutions of FDA, PCA and CCA are \textit{not sparse}.
- This often makes it \textit{difficult to interpret the results}.
- PCA/CCA: For better interpretability, \textit{few relevant features are required} that explain as much variance as possible.
  - \textit{Applications}: bio-informatics, finance, document translation etc.
- FDA: \textit{feature selection} aids generalization performance by promoting sparse solutions.
- Sparse representation $\Rightarrow$ \textit{better interpretation, better generalization and reduced computational costs}.
The variational formulation for the sparse generalized eigenvalue problem is given by

\[
\max_x \quad x^T A x \\
\text{s.t.} \quad x^T B x = 1 \\
\|x\|_0 \leq k,
\]

where \(1 \leq k \leq n\) and \(\|x\|_0 := \sum_{i=1}^n 1_{\{|x_i| \neq 0\}}\) is the cardinality of \(x\).

(2) is non-convex, NP-hard and therefore intractable.

Usually, the \(\ell_1\)-norm approximation is used for the cardinality constraint, i.e., replace \(\|x\|_0 \leq k\) by \(\|x\|_1 \leq k\).

The problem is still computationally hard.
Sparse Generalized Eigenvalue Problem

- (2) can be written as

\[
\max_x \quad x^T A x - \tilde{\rho} \|x\|_0 \\
\text{s.t.} \quad x^T B x \leq 1,
\]

where \( \tilde{\rho} \geq 0 \).

- Approximate \( \|x\|_0 \) by \( \|x\|_\varepsilon := \sum_{i=1}^n \frac{\log(1+|x_i|\varepsilon^{-1})}{\log(1+\varepsilon^{-1})} \) for sufficiently small \( \varepsilon > 0 \) as

\[
\|x\|_0 = \lim_{\varepsilon \to 0} \sum_{i=1}^n \frac{\log(1 + |x_i|\varepsilon^{-1})}{\log(1 + \varepsilon^{-1})}. \tag{4}
\]

- The approximation, \( \|x\|_\varepsilon \) can be interpreted as defining a limiting Student’s t-distribution prior over \( x \) (leading to an improper prior) given by

\[
p(x) \propto \prod_{i=1}^n \frac{1}{|x_i| + \varepsilon}
\]

and computing its negative log-likelihood.
Approximation to $||x||_0$

As $\varepsilon \to 0$, $||x||_\varepsilon \to ||x||_0$ and as $\varepsilon \to \infty$, $||x||_\varepsilon \to ||x||_1$. 
Sparse Generalized Eigenvalue Problem

- (3) reduces to the *approximate program*,

\[
\max_x x^T A x - \rho_\varepsilon \sum_{i=1}^{n} \log(|x_i| + \varepsilon)
\]

\[
\text{s.t. } x^T B x \leq 1,
\]

where \( \rho_\varepsilon := \frac{\tilde{\rho}}{\log(1+\varepsilon^{-1})} \).

- The task reduces to solving the *approximate program* in (5) with a *small value* of \( \varepsilon \).

- (5) can be written as

\[
\min_x \tau \|x\|^2 - \left( x^T (A + \tau I) x - \rho \sum_{i=1}^{n} \log(|x_i| + \varepsilon) \right)
\]

\[
\text{s.t. } x^T B x \leq 1,
\]

where \( \tau \geq \max(0, -\lambda_{\text{min}}(A)) \).

- The objective in (6) is a *difference of two convex functions*. 
Majorization-Minimization (MM)

- Suppose we want to minimize $f$ over $\Omega \subset \mathbb{R}^n$. Construct a majorization function, $g$ over $\Omega \times \Omega$ such that
  \[
  f(x) \leq g(x, y), \quad \forall x, y \in \Omega \quad \text{and} \quad f(x) = g(x, x), \quad \forall x \in \Omega.
  \]

- The majorization algorithm corresponding to $g$ updates $x$ at iteration $l$ by
  \[
  x^{(l+1)} \in \arg\min_{x \in \Omega} g(x, x^{(l)}), \quad (7)
  \]
  unless we already have
  \[
  x^{(l)} \in \arg\min_{x \in \Omega} g(x, x^{(l)}),
  \]
  in which case the algorithm stops.

- $f(x^{(l+1)}) \leq g(x^{(l+1)}, x^{(l)}) \leq g(x^{(l)}, x^{(l)}) = f(x^{(l)})$.

- MM algorithms can be thought of as a generalization of the EM algorithm.
Sparse Generalized Eigenvalue Algorithm

**Proposition**

The following function

\[
g(x, y) = \tau \|x\|_2^2 - 2x^T(A + \tau I_n)y + y^T(A + \tau I_n)y + \rho \varepsilon \sum_{i=1}^{n} \log(\varepsilon + |y_i|) \\
+ \rho \varepsilon \sum_{i=1}^{n} \frac{|x_i| - |y_i|}{|y_i| + \varepsilon},
\]

majorizes the objective function in (6).

By following the minimization step in (7) with \(g\) as in (8), the *sparse GEV algorithm* is obtained as

\[
x^{(l+1)} = \arg \min_x \tau \|x\|_2^2 - 2x^T(A + \tau I_n)x^{(l)} + \rho \varepsilon \sum_{i=1}^{n} \frac{|x_i|}{|x_i^{(l)}| + \varepsilon} \\
\text{s.t. } x^T B x \leq 1,
\]

which is a sequence of quadratically constrained quadratic programs (QCQPs).
Sparse Generalized Eigenvalue Program

- (9) can also be written as

\[
x^{(l+1)} = \arg \min_x \| x - (\tau^{-1} A + I_n)x^{(l)} \|^2_2 + \frac{\rho}{\tau} \| W^{(l)}x \|_1
\]

s.t. \( x^T B x \leq 1, \)

where \( w_i^{(l)} := \frac{1}{|x_i^{(l)}| + \epsilon}, \ w^{(l)} := (w_1^{(l)}, \ldots, w_n^{(l)}) \) and \( W^{(l)} := \text{diag}(w^{(l)}). \)

- (10) is very similar to \textit{LASSO} [Tibshirani, 1996] except for the \textit{weighted} \( \ell_1 \)-norm penalty and the quadratic constraint.

- When \( A \succeq 0, \ B = I_n \) and \( \tau = 0 \), (9) reduces to a very simple iterative rule:

\[
x_i^{(l+1)} = \frac{\left[ |(Ax^{(l)})_i| - \frac{\rho \epsilon}{2} w_i^{(l)} \right]_+ \text{sign}((Ax^{(l)})_i)}{\sqrt{\sum_{i=1}^n \left[ |(Ax^{(l)})_i| - \frac{\rho \epsilon}{2} w_i^{(l)} \right]^2}}, \ \forall \ i,
\]

where \([a]_+ := \max(0, a)\), which we call as DC-PCA.
Convergence Analysis

Theorem

Let \( \{ x^{(l)} \}_{l=0}^{\infty} \) be any sequence generated by the sparse GEV algorithm in (9). Then, all the limit points of \( \{ x^{(l)} \}_{l=0}^{\infty} \) are stationary points of the program in (5),

\[
\rho \varepsilon \sum_{i=1}^{n} \log(\varepsilon + |x_i^{(l)}|) - [x^{(l)}]^{T}Ax^{(l)} \rightarrow \rho \varepsilon \sum_{i=1}^{n} \log(\varepsilon + |x_i^{*}|) - [x^{*}]^{T}Ax^{*} := L^*,
\]

for some stationary point \( x^* \), \( \|x^{(l+1)} - x^{(l)}\| \rightarrow 0 \), and either \( \{x^{(l)}\}_{l=0}^{\infty} \) converges or the set of limit points of \( \{x^{(l)}\}_{l=0}^{\infty} \) is a connected and compact subset of \( \mathcal{S}(L^*) \), where

\[
\mathcal{S}(a) := \{ x \in \mathcal{S} : x^{T}Ax - \rho \varepsilon \sum_{i=1}^{n} \log(\varepsilon + |x_i|) = -a \}
\]

and \( \mathcal{S} \) is the set of stationary points of (5). If \( \mathcal{S}(L^*) \) is finite, then any sequence \( \{x^{(l)}\}_{l=0}^{\infty} \) generated by (9) converges to some \( x^* \) in \( \mathcal{S}(L^*) \).
Convergence Analysis

Corollary
Let $\hat{\rho} = 0$ and $\lambda_{\text{max}}(A, B) > 0$. Then, any sequence $\{x^{(l)}\}_{l=0}^\infty$ generated by (9) converges to some $x^*$ such that $\lambda_{\text{max}}(A, B) = [x^*]^T Ax^*$ and $[x^*]^T Bx^* = 1$.

- Local and global solutions are the same for $\rho = 0$.

Corollary
Let $A \succeq 0$, $\tau = 0$ and $\hat{\rho} = 0$. Then, any sequence $\{x^{(l)}\}_{l=0}^\infty$ generated by the following algorithm

$$x^{(l+1)} = \frac{B^{-1}Ax^{(l)}}{\sqrt{[x^{(l)}]^T AB^{-1}Ax^{(l)}}}$$

converges to some $x^*$ such that $\lambda_{\text{max}}(A, B) = [x^*]^T Ax^*$ and $[x^*]^T Bx^* = 1$.

- With $B = I_n$, (12) reduces to the power method for computing $\lambda_{\text{max}}(A)$. 

Applications: Sparse PCA

- **Sparse PCA algorithms**: Proposed (DC-PCA), SDP relaxation (DSPCA [d’Aspremont et al., 2005]), greedy approach (GSPCA [Moghaddam et al., 2007]), regression based approach (SPCA [Zou et al., 2006]) and generalized power method (GPower$_{\ell_0}$ [Journée et al., 2008]).

- **Pit props data** [Jeffers, 1967]
  - A benchmark data to test sparse PCA algorithms.
  - 180 observations and 13 measured variables.
  - 6 principal directions are considered as they capture 87% of the total variance.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Sparsity pattern</th>
<th>Cumulative cardinality</th>
<th>Cumulative variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPCA</td>
<td>(7,4,4,1,1,1)</td>
<td>18</td>
<td>75.8%</td>
</tr>
<tr>
<td>DSPCA</td>
<td>(6,2,3,1,1,1)</td>
<td>14</td>
<td>75.5%</td>
</tr>
<tr>
<td>GSPCA</td>
<td>(6,2,2,1,1,1)</td>
<td>13</td>
<td>77.1%</td>
</tr>
<tr>
<td>GPower$_{\ell_0}$</td>
<td>(6,2,2,1,1,1)</td>
<td>13</td>
<td>77.1%</td>
</tr>
<tr>
<td>DC-PCA</td>
<td>(6,2,2,1,1,1)</td>
<td>13</td>
<td>77.1%</td>
</tr>
</tbody>
</table>
Figure: (a) cumulative variance and (b) cumulative cardinality for the first 6 sparse PCs; (c) proportion of explained variance (PEV) vs. cardinality for the first sparse PC; (d) dependence of sparsity and PEV on $\tilde{\rho}$ for the first sparse PC computed with DC-PCA.
Gene Datasets

*Table:* Gene expression datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Samples ($p$)</th>
<th>Genes ($n$)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colon cancer</td>
<td>62</td>
<td>2000</td>
<td>[Alon et al., 1999]</td>
</tr>
<tr>
<td>Leukemia</td>
<td>38</td>
<td>7129</td>
<td>[Golub et al., 1999]</td>
</tr>
<tr>
<td>Ramaswamy</td>
<td>127</td>
<td>16063</td>
<td>[Ramaswamy et al., 2001]</td>
</tr>
</tbody>
</table>

*Table:* Computation time (in seconds) to obtain the first sparse PC, averaged over cardinalities ranging from 1 to $n$, for the Colon cancer, Leukemia and Ramaswamy datasets.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Colon cancer</th>
<th>Leukemia</th>
<th>Ramaswamy</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2000$</td>
<td>2.057</td>
<td>3.548</td>
<td>38.731</td>
</tr>
<tr>
<td>$7129$</td>
<td>0.182</td>
<td>0.223</td>
<td>2.337</td>
</tr>
<tr>
<td>$16063$</td>
<td>0.034</td>
<td>0.156</td>
<td>0.547</td>
</tr>
</tbody>
</table>
**Gene Datasets**

*Figure:* Trade-off curves between explained variance and cardinality for (a) Colon cancer, (b) Leukemia and (c) Ramaswamy datasets. The proportion of variance explained is computed on the first sparse principal component.
Scalability

- **Complexity**
  - DC-PCA, GPower$_{\ell_0}$: $O(mn^2)$, where $m$ is the number of iterations before convergence.
  - SPCA: $O(mn^3)$
  - GSPCA: $O(n^4)$
  - DSPCA: $O(n^4 \sqrt{\log n})$

- Randomly chosen problems of size $n$ ranging from 10 to 10000.
- Linux 3 GHz, 4 GB RAM workstation.

*Figure:* Average computation time (seconds) for the first sparse PC of $A$ vs. problem size, $n$, over 100 randomly generated matrices $A$. 
References


