Semidefinite Optimization
with Applications in Sparse Multivariate Statistics

Alexandre d'Aspremont
ORFE, Princeton University

Joint work with L. El Ghaoui, M. Jordan, V. Krishnamurthy, G. Lanckriet, R. Luss and Nathan Srebro.

Support from NSF and Google.

Introduction

Semidefinite Programming:

- Essentially: linear programming over positive semidefinite matrices.
- Sounds very specialized but has applications everywhere (often non-obvious)...
- One example here: convex relaxations of combinatorial problems.

Sparse Multivariate Statistics:

- Sparse variants of PCA, SVD, etc are combinatorial problems.
- Efficient **relaxations** using semidefinite programming.
- Solve realistically large problems.

Linear Programming

A **linear program** (LP) is written:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \succeq 0 \end{array}$$

its dual is another LP:

$$\begin{array}{ll} \text{maximize} & b^T x \\ \text{subject to} & A^T y \preceq c \end{array}$$

- Here, $x \succeq 0$ means that the **vector** $x \in \mathbf{R}^n$ has **nonnegative** coefficients.
- First solved using the simplex algorithm (exponential complexity).
- Using interior point methods, complexity is $O(n^{3.5})$.

Semidefinite Programming

A **semidefinite program** (SDP) is written:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(CX) \\ \text{subject to} & \mathbf{Tr}(A_i X) = b_i, \quad i = 1, \ldots, m \\ & X \succeq 0 \end{array}$$

its dual is:

$$\begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & \sum_i y_i A_i \preceq C \end{array}$$

- Here, $X \succeq 0$ means that the **matrix** $X \in \mathbf{S}_n$ is **positive semidefinite**.
- Nesterov & Nemirovskii (1994) extended the complexity analysis of interior point methods used for solving LPs to semidefinite programs (and others).
- Complexity in $O(n^{4.5})$ when $m \sim n$ (see Ben-Tal & Nemirovski (2001)), harder to exploit problem structure such as sparsity, low-rank matrices, etc.

Outline

- **Two classic relaxation tricks**
  - Semidefinite relaxations and the lifting technique
  - The l_1 heuristic
- Applications
  - Covariance selection
  - Sparse PCA, SVD
  - Sparse nonnegative matrix factorization
- Solving large-scale semidefinite programs
  - First-order methods
  - Numerical performance

Semidefinite relaxations

Easy & Hard Problems. . .

Classical view on complexity:

- **linear** is easy
- **nonlinear** is hard(er)

Correct view:

- **convex** is easy
- **nonconvex** is hard(er)

Convex Optimization

Problem format:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_1(x) \leq 0, \ldots, f_m(x) \leq 0 \end{array}$$

where $x \in \mathbf{R}^n$ is the optimization variable and $f_i : \mathbf{R}^n \to \mathbf{R}$ are **convex**.

- includes LS, LP, QP, and **many others**
- like LS, LP, and QP, convex problems are **fundamentally tractable** (cf. ellipsoid method)

Nonconvexity makes problems **essentially untractable**...

- Sometimes the result of bad problem formulation
- However, often arises because of some natural limitation: fixed transaction costs, binary communications, ...

We can use convex optimization results to find bounds on the optimal value an approximate solutions by **relaxation**.

Basic Problem

- We focus here on a specific class of problems: Quadratically Constrained Quadratic Programs (QCQP).
- Vast range of applications...

A **QCQP** can be written:

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m \end{array}$$

- If all P_i are positive semidefinite, this is a convex problem: easy.
- Here, we suppose at least **one P_i not p.s.d.**

Example: Partitioning Problem

Two-way partitioning problem:

$$\begin{array}{ll} \text{minimize} & x^T W x \\ \text{subject to} & x_i^2 = 1, \quad i = 1, \ldots, n \end{array}$$

where $W \in \mathbf{S}_n$, with $W_{ii} = 0$. A QCQP in the variable $x \in \mathbf{R}^n$.

- A feasible x corresponds to the partition

$$\{1, \ldots, n\} \;=\; \{i \mid x_i = -1\} \;\cup\; \{i \mid x_i = 1\}.$$

- The matrix coefficient W_{ij} can be interpreted as the cost of having the elements i and j in the same partition.
- The objective is to find the partition with least total cost.
- Classic particular instance: MAXCUT $(W_{ij} \geq 0)$.

Semidefinite Relaxation

The original QCQP:

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m \end{array}$$

can be rewritten:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(XP_0) + q_0^T x + r_0 \\ \text{subject to} & \mathbf{Tr}(XP_i) + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m \\ & X = xx^T. \end{array}$$

This is the **same problem** (lifted in $\mathbf{S}_n$).

Semidefinite Relaxation

We can replace $X = xx^T$ by $X \succeq xx^T$, $\;\mathbf{Rank}(X) = 1$, so this is again:

$$\begin{array}{ll}\mbox{minimize} & \mathbf{Tr}(XP_0) + q_0^T x + r_0\\ \mbox{subject to} & \mathbf{Tr}(XP_i) + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m\\ & X \succeq xx^T, \;\mathbf{Rank}(X) = 1\end{array}$$

The constraint $X \succeq xx^T$ is a Schur complement constraint and is convex. The only remaining nonconvex constraint is now $\mathbf{Rank}(X) = 1$. We simply drop it and solve:

$$\begin{array}{ll}\mbox{minimize} & \mathbf{Tr}(XP_0) + q_0^T x + r_0\\ \mbox{subject to} & \mathbf{Tr}(XP_i) + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m\\ & \left[\begin{array}{cc} X & x^T\\ x & 1\end{array}\right] \succeq 0\end{array}$$

This is a **semidefinite program** in $X \in \mathbf{S}_n$.

Semidefinite Relaxation

The original QCQP:

$$\begin{array}{ll} \mbox{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \mbox{subject to} & x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m \end{array}$$

was relaxed as:

$$\begin{array}{ll} \mbox{minimize} & \mathbf{Tr}(XP_0) + q_0^T x + r_0 \\ \mbox{subject to} & \mathbf{Tr}(XP_i) + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m \\ & \left[\begin{array}{cc} X & x^T \\ x & 1 \end{array}\right] \succeq 0 \end{array}$$

- The relaxed problem is convex and can be solved efficiently.
- The optimal value of the SDP is a lower bound on the solution of the original problem.

Semidefinite Relaxation: Partitioning

The partitioning problem defined was a QCQP:

$$\begin{array}{ll} \mbox{minimize} & x^TWx \\ \mbox{subject to} & x_i^2 = 1, \quad i = 1,\ldots,n \end{array}$$

There are only quadratic terms, so the variable x disappears from the relaxation, which becomes:

$$\begin{array}{ll} \mbox{minimize} & \mathbf{Tr}(WX) \\ \mbox{subject to} & X \succeq 0 \\ & X_{ii} = 1, \quad i = 1,\ldots,n \end{array}$$

- These relaxations only provide a lower bound on the optimal value.
- If $\mathbf{Rank}(X) = 1$ at the optimum, $X = xx^T$ and the relaxation is tight.
- How can we compute good feasible points otherwise?
- One solution: take the dominant eigenvector of X and project it on $\{-1, 1\}$.

Randomization

The original QCQP:

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m \end{array}$$

was relaxed into:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(XP_0) + q_0^T x + r_0 \\ \text{subject to} & \mathbf{Tr}(XP_i) + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m \\ & X \succeq xx^T \end{array}$$

- The last constraint means $X - xx^T$ is a **covariance** matrix...
- Pick y as a Gaussian variable with $y \sim \mathcal{N}(x, X - xx^T)$, y will solve the QCQP "on average" over this distribution, in other words:

$$\begin{array}{ll} \text{minimize} & \mathbf{E}[y^T P_0 y + q_0^T y + r_0] \\ \text{subject to} & \mathbf{E}[y^T P_i y + q_i^T y + r_i] \leq 0, \quad i = 1, \ldots, m \end{array}$$

- A good feasible point can then be obtained by **sampling** enough x...

Outline

- Two classic relaxation tricks
  - Semidefinite relaxations and the lifting technique
  - **The l_1 heuristic**
- Applications
  - Covariance selection
  - Sparse PCA, SVD
  - Sparse nonnegative matrix factorization
- Solving large-scale semidefinite programs
  - First-order methods
  - Numerical performance

The l_1 heuristic

Start from a linear system:

$$Ax = b$$

with $A \in \mathbf{R}^{m \times n}$ where $m < n$. We look for a **sparse** solution:

$$\begin{array}{ll} \mbox{minimize} & \mathbf{Card}(x) \\ \mbox{subject to} & Ax = b. \end{array}$$

If the solution set is bounded, this can be formulated as a Mixed Integer Linear Program:

$$\begin{array}{ll} \mbox{minimize} & \mathbf{1}^T u \\ \mbox{subject to} & Ax = b \\ & |x| \preceq Bu \\ & u \in \{0,1\}^n. \end{array}$$

This is a hard problem. . .

l_1 relaxation

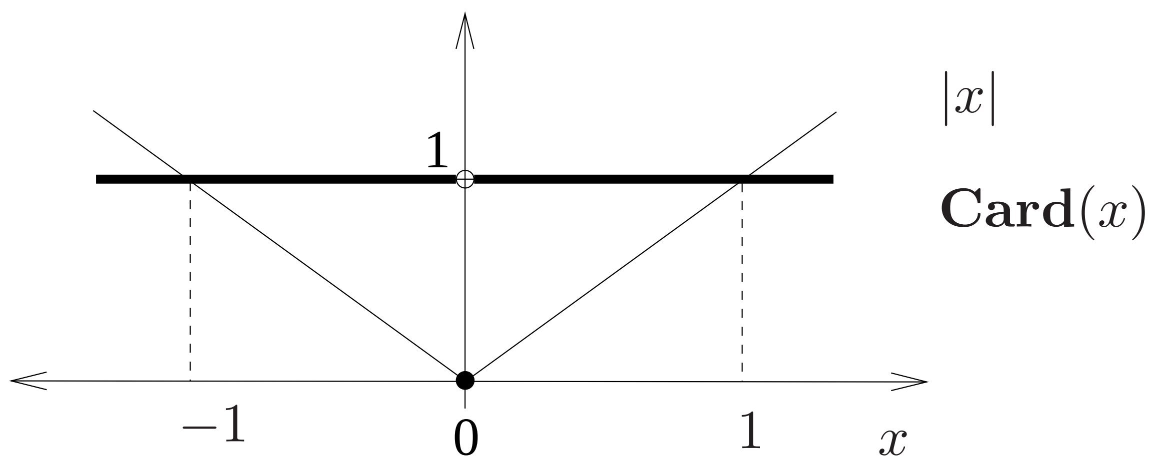
Assuming $|x| \leq 1$, we can replace:

$$\mathbf{Card}(x) = \sum_{i=1}^{n} 1_{\{x_i \neq 0\}}$$

with

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

Graphically, assuming $x \in [-1, 1]$ this is:



The l_1 norm is the **largest convex lower bound** on $\mathbf{Card}(x)$ in $[-1, 1]$.

l_1 relaxation

$$\begin{array}{ll} \text{minimize} & \mathbf{Card}(x) \\ \text{subject to} & Ax = b \end{array} \quad \textbf{becomes} \quad \begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = b \end{array}$$

- The relaxed problem is a **linear program**.
- This trick can be used for other problems (cf. **minimum rank** result from Fazel, Hindi & Boyd (2001)).
- Candès & Tao (2005) or Donoho & Tanner (2005) show that if there is a sufficiently sparse solution, it is optimal and the relaxation is **tight**. (This result only works in the linear case).

l_1 relaxation

The original problem in MILP format:

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T u \\ \text{subject to} & Ax = b \\ & |x| \preceq Bu \\ & u \in \{0,1\}^n, \end{array}$$

can be reformulated as a (nonconvex) **QCQP**:

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T u \\ \text{subject to} & Ax = b \\ & -x \preceq Bu, \; x \preceq Bu \\ & u_i^2 = u_i, \quad i = 1, \ldots, n. \end{array}$$

- We could also formulate a **semidefinite relaxation**.
- Lemaréchal & Oustry (1999) show that this is **equivalent** to relaxing $u \in \{0,1\}^n$ as $u \in [0,1]^n$, which is exactly the l_1 heuristic.

Outline

- Two classic relaxation tricks
  - Semidefinite relaxations and the lifting technique
  - The l_1 heuristic
- **Applications**
  - Covariance selection
  - Sparse PCA, SVD
  - Sparse nonnegative matrix factorization
- Solving large-scale semidefinite programs
  - First-order methods
  - Numerical performance

Covariance Selection

We estimate a **sample covariance matrix** Σ from empirical data...

- Objective: infer **dependence** relationships between variables.
- We want this information to be as **sparse** as possible.
- Basic solution: look at the magnitude of the covariance coefficients:

$$|\Sigma_{ij}| > \beta \quad \Leftrightarrow \quad \text{variables } i \text{ and } j \text{ are related,}$$

  and simply threshold smaller coefficients to zero. (not always psd.)

We can do better...

Covariance Selection

Following Dempster (1972), look for zeros in the **inverse** covariance matrix:

- **Parsimony**. Suppose that we are estimating a Gaussian density:

$$f(x,\Sigma) = \left(\frac{1}{2\pi}\right)^{\frac{p}{2}} \left(\frac{1}{\det \Sigma}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right),$$

  a sparse inverse matrix Σ^{-1} corresponds to a **sparse representation** of the density f as a member of an exponential family of distributions:

$$f(x,\Sigma) = \exp(\alpha_0 + t(x) + \alpha_{11}t_{11}(x) + \ldots + \alpha_{rs}t_{rs}(x))$$

  with here $t_{ij}(x) = x_i x_j$ and $\alpha_{ij} = \Sigma^{-1}_{ij}$.

- Dempster (1972) calls Σ^{-1}_{ij} a **concentration** coefficient.

There is more. . .

Covariance Selection

Conditional independence:

- Suppose X, Y, Z have are jointly normal with covariance matrix Σ, with

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

  where $\Sigma_{11} \in \mathbf{R}^{2\times 2}$ and $\Sigma_{22} \in \mathbf{R}$.

- Conditioned on Z, X, Y are still normally distributed with covariance matrix C given by:

$$C = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = \left(\left(\Sigma^{-1}\right)_{11}\right)^{-1}$$

- So X and Y are **conditionally independent** iff $\left(\Sigma^{-1}\right)_{11}$ is diagonal, which is also:

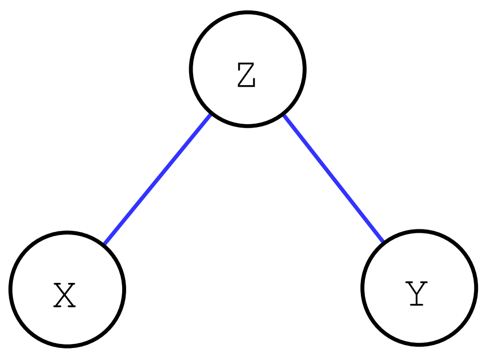
$$\Sigma_{xy}^{-1} = 0$$

Covariance Selection

- Suppose we have iid noise $\epsilon_i \sim \mathcal{N}(0,1)$ and the following linear model:

$$\begin{aligned} x &= z + \epsilon_1 \\ y &= z + \epsilon_2 \\ z &= \epsilon_3 \end{aligned}$$

- Graphically, this is:

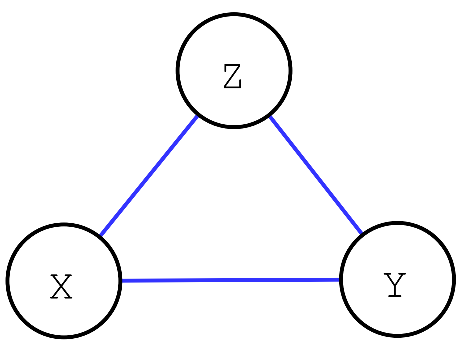


Covariance Selection

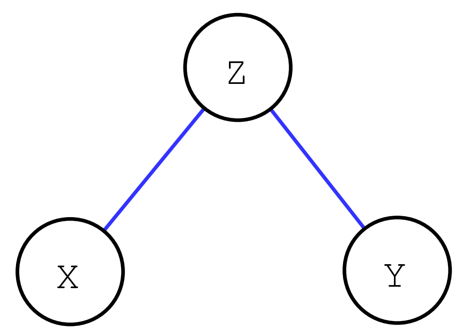
- The covariance matrix and inverse covariance are given by:

$$\Sigma = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad \Sigma^{-1} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 3 \end{pmatrix}$$

- The inverse covariance matrix has Σ^{-1}_{12} clearly showing that the variables x and y are independent conditioned on z.
- Graphically, this is again:

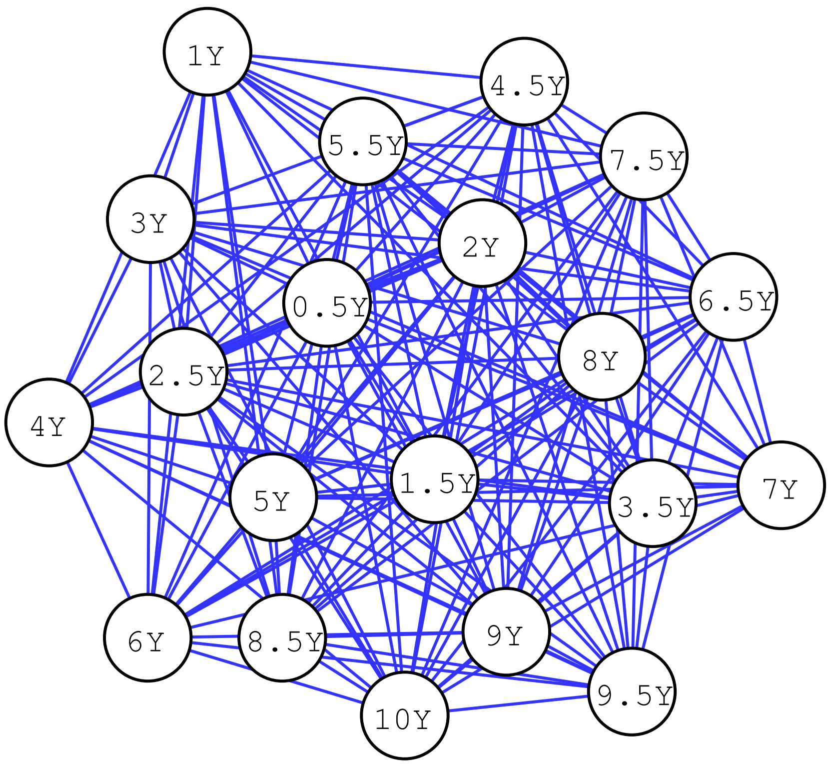


versus

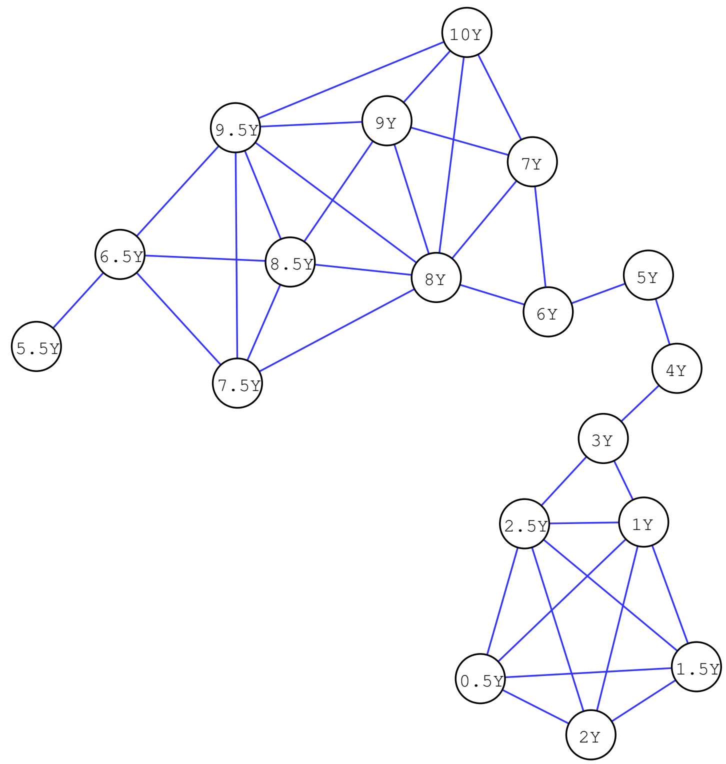


Covariance Selection

On a slightly larger scale. . .



Before



After

Applications & Related Work

- **Gene expression data**. The sample data is composed of gene expression vectors and we want to isolate links in the expression of various genes. See Dobra, Hans, Jones, Nevins, Yao & West (2004), Dobra & West (2004) for example.
- **Speech Recognition**. See Bilmes (1999), Bilmes (2000) or Chen & Gopinath (1999).
- **Finance**. Covariance estimation.
- Related work by Dahl, Roychowdhury & Vandenberghe (2005): interior point methods for large, sparse MLE.

Maximum Likelihood Estimation

- We can estimate Σ by solving the following maximum likelihood problem:

$$\max_{X \in \mathbf{S}^n} \log\det X - \mathbf{Tr}(SX)$$

- This problem is convex, has an explicit answer $\Sigma = S^{-1}$ if $S \succ 0$.
- Problem here: how do we make Σ^{-1} **sparse**?
- In other words, how do we efficiently choose I and J?
- Solution: penalize the MLE.

AIC and BIC

Original solution in Akaike (1973), **penalize** the likelihood function:

$$\max_{X \in \mathbf{S}^n} \log\det X - \mathbf{Tr}(SX) - \rho\,\mathbf{Card}(X)$$

where $\mathbf{Card}(X)$ is the number of nonzero elements in X.

- Set $\rho = 2/(m+1)$ for **AIC** and $\rho = \log(m+1)/(m+1)$ for **BIC**.
- We can form a **convex relaxation** of AIC or BIC penalized MLE by replacing $\mathbf{Card}(X)$ by $\|X\|_1 = \sum_{ij} |X_{ij}|$ to solve:

$$\max_{X \in \mathbf{S}^n} \log\det X - \mathbf{Tr}(SX) - \rho\|X\|_1$$

Again, the classic l_1 heuristic: $\|X\|_1$ is a **convex lower bound** on $\mathbf{Card}(X)$.

Robustness

- This penalized MLE problem can be rewritten:

$$\max_{X \in \mathbf{S}^n} \min_{|U_{ij}| \leq \rho} \log\det X - \mathbf{Tr}((S+U)X)$$

- This can be interpreted as a **robust MLE** problem with componentwise noise of magnitude ρ on the elements of S.

- The relaxed **sparsity** requirement is equivalent to a **robustification**.

Outline

- Two classic relaxation tricks
  - Semidefinite relaxations and the lifting technique
  - The l_1 heuristic
- Applications
  - Covariance selection
  - **Sparse PCA, SVD**
  - Sparse nonnegative matrix factorization
- Solving large-scale semidefinite programs
  - First-order methods
  - Numerical performance

Sparse Principal Component Analysis

Principal Component Analysis (PCA): classic tool in multivariate data analysis

- Input: a **covariance** matrix A
- Output: a sequence of **factors** ranked by variance
- Each factor is a linear combination of the problem variables

Typical use: **dimensionality reduction**.

Numerically, just an eigenvalue decomposition of the covariance matrix:

$$A = \sum_{i=1}^{n} \lambda_i x_i x_i^T$$

Sparse Principal Component Analysis

Computing factors amounts to solving:

$$\begin{array}{ll} \text{maximize} & x^T A x \\ \text{subject to} & \|x\|_2 = 1. \end{array}$$

This problem is **easy**, its solution is again $\lambda^{\max}(A)$ at x_1. Here however, we want a little bit more...

We look for a **sparse** solution and solve instead:

$$\begin{array}{ll} \text{maximize} & x^T A x \\ \text{subject to} & \|x\|_2 = 1 \\ & \mathbf{Card}(x) \leq k, \end{array}$$

where $\mathbf{Card}(x)$ denotes the cardinality (number of non-zero elements) of x. This is non-convex and **numerically hard**.

Related literature

Previous work:

- Cadima & Jolliffe (1995): the loadings with small absolute value are thresholded to zero.
- A non-convex method called SCoTLASS by Jolliffe, Trendafilov & Uddin (2003). (Same problem formulation)
- Zou, Hastie & Tibshirani (2004): a regression based technique called SPCA. Based on a representation of PCA as a regression problem. Sparsity is obtained using the LASSO Tibshirani (1996) a l_1 norm penalty.

Performance:

- These methods are either very suboptimal (thresholding) or lead to **nonconvex** optimization problems (SPCA).
- Regression: works for very **large scale** examples.

Semidefinite relaxation

Start from

$$\begin{array}{ll} \text{max} & x^T A x \\ \text{subject to} & \|x\|_2 = 1 \\ & \mathbf{Card}(x) \leq k, \end{array}$$

Let $X = xx^T$, and write everything in terms of the matrix X:

$$\begin{array}{ll} \text{max} & \mathbf{Tr}(AX) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & \mathbf{Card}(X) \leq k^2 \\ & X = xx^T, \end{array}$$

Replace $X = xx^T$ by the equivalent $X \succeq 0, \ \mathbf{Rank}(X) = 1$:

$$\begin{array}{ll} \text{max} & \mathbf{Tr}(AX) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & \mathbf{Card}(X) \leq k^2 \\ & X \succeq 0, \ \mathbf{Rank}(X) = 1, \end{array}$$

again, this is the **same problem**.

Semidefinite relaxation

Numerically, this is still **hard**:

- The $\mathbf{Card}(X) \leq k^2$ is still non-convex
- So is the constraint $\mathbf{Rank}(X) = 1$

However, we have made **some progress**:

- The objective $\mathbf{Tr}(AX)$ is now **linear** in X
- The (non-convex) constraint $\|x\|_2 = 1$ became a **linear** constraint $\mathbf{Tr}(X) = 1$.

We still need to relax the two non-convex constraints above:

- If $u \in \mathbf{R}^p$, $\mathbf{Card}(u) = q$ implies $\|u\|_1 \leq \sqrt{q}\|u\|_2$. So we can replace $\mathbf{Card}(X) \leq k^2$ by the weaker (but **convex**): $\mathbf{1}^T|X|\mathbf{1} \leq k$
- Simply drop the rank constraint

Semidefinite relaxation

Semidefinite relaxation combined with l_1 **heuristic**:

$$\begin{array}{ll} \text{max} & x^T A x \\ \text{subject to} & \|x\|_2 = 1 \\ & \mathbf{Card}(x) \leq k, \end{array} \qquad \textbf{becomes} \qquad \begin{array}{ll} \text{max} & \mathbf{Tr}(AX) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & \mathbf{1}^T |X| \mathbf{1} \leq k \\ & X \succeq 0, \end{array}$$

- This is a **semidefinite program** in the variable $X \in \mathbf{S}^n$, polynomial complexity. . .
- Small problem instances can be solved using **SEDUMI** by Sturm (1999) or **SDPT3** by Toh, Todd & Tutuncu (1999).
- This semidefinite program has $O(n^2)$ dense constraints on the matrix, we want to solve large problems $n \sim 10^3$.

Can't use interior point methods. . .

Robustness & Tightness

Robustness. The penalized problem can be written:

$$\min_{\{|U_{ij}|\leq\rho\}} \quad \lambda^{\max}(A+U)$$

Natural interpretation: **robust** maximum eigenvalue problem with componentwise noise of magnitude ρ on the coefficients of the matrix A.

Tightness. The KKT optimality conditions are here:

$$\begin{cases} (A+U)X = \lambda^{\max}(A+U)X \\ U \circ X = \rho |X| \\ \mathbf{Tr}(X) = 1,\ X \succeq 0 \\ |U_{ij}| \leq \rho, \quad i,j = 1,\ldots,n. \end{cases}$$

The first order condition means that if $\lambda^{\max}(A+U)$ is simple, $\mathbf{Rank}(X) = 1$ so the relaxation is **tight**: the solution to the relaxed problem is also a global optimum for the original combinatorial problem.

Sparse Singular Value Decomposition

A similar reasoning involves a **non-square** $m \times n$ matrix A, and the problem

$$\begin{array}{ll}\text{max} & u^T A v \\ \text{subject to} & \|u\|_2 = \|v\|_2 = 1 \\ & \mathbf{Card}(u) \leq k_1, \;\; \mathbf{Card}(v) \leq k_2,\end{array}$$

in the variables $(u,v) \in \mathbf{R}^m \times \mathbf{R}^n$ where $k_1 \leq m$, $k_2 \leq n$ are fixed. This is relaxed as:

$$\begin{array}{ll}\text{max} & \mathbf{Tr}(A^T X^{12}) \\ \text{subject to} & \mathbf{1}^T |X^{ii}| \mathbf{1} \leq k_i, \quad i = 1,2 \\ & \mathbf{1}^T |X^{12}| \mathbf{1} \leq \sqrt{k_1 k_2} \\ & X \succeq 0, \;\; \mathbf{Tr}(X^{ii}) = 1\end{array}$$

in the variable $X \in \mathbf{S}^{m+n}$ with blocks X^{ij} for $i,j = 1,2$, using the fact that the eigenvalues of the matrix:

$$\left[\begin{array}{cc} 0 & A \\ A^T & 0 \end{array}\right]$$

are $\{\sigma_i, \ldots, -\sigma_i, \ldots\}$ where σ are the singular values of the matrix $A \in \mathbf{R}^{m \times n}$.

Nonnegative Matrix Factorization

Direct extension of sparse PCA result. . . Solving

$$\begin{array}{ll} \text{max} & u^T A v \\ \text{subject to} & \|u\|_2 = \|v\|_2 = 1 \\ & \mathbf{Card}(u) \leq k_1, \;\; \mathbf{Card}(v) \leq k_2, \end{array}$$

also solves:

$$\begin{array}{ll} \text{min} & \|A - uv^T\|_F \\ \text{subject to} & \mathbf{Card}(u) \leq k_1 \\ & \mathbf{Card}(v) \leq k_2, \end{array}$$

So, by adding constraints on u and v we can use the previous result to form a relaxation for the **Nonnegative Matrix Factorization** problem:

$$\begin{array}{ll} \text{max} & \mathbf{Tr}(A^T X^{12}) \\ \text{subject to} & \mathbf{1}^T |X^{ii}| \mathbf{1} \leq k_i, \quad i = 1, 2 \\ & \mathbf{1}^T |X^{12}| \mathbf{1} \leq \sqrt{k_1 k_2} \\ & X \succeq 0, \;\; \mathbf{Tr}(X^{ii}) = 1 \\ & X_{ij} \geq 0, \end{array}$$

Caveat: only works with rank one factorization. . .

Outline

- Two classic relaxation tricks
  - Semidefinite relaxations and the lifting technique
  - The l_1 heuristic
- Applications
  - Covariance selection
  - Sparse PCA, SVD
  - Sparse nonnegative matrix factorization
- **Solving large-scale semidefinite programs**
  - First-order methods
  - Numerical performance

Outline

Most of our problems are **dense**, with $n \sim 10^3$.

Solver options:

- Interior point methods fail beyond $n \sim 400$.
- Projected subgradient: extremely slow.
- Bundle method (see Helmberg & Rendl (2000)): a bit faster, but can't take advantage of box-like structure of feasible set. Convergence in $O(1/\epsilon^2)$.

First order algorithm

Complexity options. . .

$O(n)$ $O(n)$ $O(n^2)$

Memory

$O(1/\epsilon^2)$ $O(1/\epsilon)$ $O(\log(1/\epsilon))$

First-order Smooth Newton IP **Complexity**

First order algorithm

Here, we can exploit problem structure

- Our problems here have **min-max** structure. For sparse PCA:

$$\min_{|U_{ij}|\leq\rho} \lambda^{\max}(A+U) \;=\; \min_{|U_{ij}|\leq\rho} \max_{X\in\mathbf{S}^n} \mathbf{Tr}((A+U)X)$$

- This min-max structure means that we can use prox function algorithms by Nesterov (2005) (see also Nemirovski (2004)) to solve large, dense problem instances.

First order algorithm

Solve

$$\min_{x \in Q_1} f(x)$$

- Starts from a particular **min-max model** on the problem:

$$f(x) = \hat{f}(x) + \max_u \{\langle Tx, u\rangle - \hat{\phi}(u) \ : \ u \in Q_2\}$$

- assuming that:
  - f is defined over a compact convex set $Q_1 \subset \mathbf{R}^n$
  - $\hat{f}(x)$ is convex, differentiable and has a Lipschitz continuous gradient with constant $M \geq 0$
  - T is a linear operator: $T \in \mathbf{R}^{n \times n}$
  - $\hat{\phi}(u)$ is a continuous convex function over some compact set $Q_2 \subset \mathbf{R}^n$.

First order algorithm

If problem has min-max model, **two steps**:

- **Regularization**. Add strongly convex penalty inside the min-max representation to produce an ϵ-approximation of f with Lipschitz continuous gradient (generalized Moreau-Yosida regularization step, see Lemaréchal & Sagastizábal (1997) for example).
- **Optimal first order minimization**. Use optimal first order scheme for Lipschitz continuous functions detailed in Nesterov (1983) to the solve the regularized problem.

Benefits:

- For fixed problem size, the number of iterations required to get an ϵ solution is given by $O\left(1/\epsilon\right)$ compared to $O\left(1/\epsilon^2\right)$ for generic first-order methods.
- Low memory requirements: change in **granularity** of the solver: larger number of cheaper iterations.

Caveat: Only efficient if the subproblems involved in these steps can be solved explicitly or extremely efficiently. . .

First order algorithm

Regularization. We can find a uniform ϵ-approximation to $\lambda^{\max}(X)$ with Lipschitz continuous gradient. Let $\mu > 0$ and $X \in \mathbf{S}_n$, we define:

$$f_\mu(X) = \mu \log \mathbf{Tr}\left(\exp\left(\frac{X}{\mu}\right)\right)$$

which requires computing a matrix exponential at a numerical cost of $O(n^3)$. We then have:

$$\lambda^{\max}(X) \leq f_\mu(X) \leq \lambda^{\max}(X) + \mu \log n,$$

so if we set $\mu = \epsilon / \log n$, $f_\mu(X)$ becomes a **uniform ϵ-approximation** of $\lambda^{\max}(X)$ and $f_\mu(X)$ has a **Lipschitz continuous gradient** with constant:

$$L = \frac{1}{\mu} = \frac{\log n}{\epsilon}.$$

The gradient $\nabla f_\mu(X)$ can also be computed explicitly as:

$$\exp\left(\frac{X - \lambda^{\max}(X)\mathbf{I}}{\mu}\right) / \mathbf{Tr}\left(\exp\left(\frac{X - \lambda^{\max}(X)\mathbf{I}}{\mu}\right)\right)$$

using the same matrix exponential.

First order algorithm

Optimal first-order minimization. The minimization algorithm in Nesterov (1983) then involves the following steps:

Choose $\epsilon > 0$ and set $X_0 = \beta I_n$, **For** $k = 0, \ldots, N(\epsilon)$ **do**

1. Compute $\nabla f_\epsilon(X_k)$
2. Find $Y_k = \arg\min_Y \left\{\mathbf{Tr}(\nabla f_\epsilon(X_k)(Y - X_k)) + \frac{1}{2} L_\epsilon \|Y - X_k\|_F^2 \; : \; Y \in \mathcal{Q}_1\right\}$.
3. Find
   $Z_k = \arg\min_X \left\{ L_\epsilon \beta^2 d_1(X) + \sum_{i=0}^{k} \frac{i+1}{2} \mathbf{Tr}(\nabla f_\epsilon(X_i)(X - X_i)) \; : \; X \in \mathcal{Q}_1 \right\}$.
4. Update $X_k = \frac{2}{k+3} Z_k + \frac{k+1}{k+3} Y_k$.
5. Test if gap less than target precision.

- **Step 1** requires computing a matrix exponential.
- **Steps 2 and 3** are both Euclidean projections on $Q_1 = \{U : \; |U_{ij} \leq \rho\}$.

First order algorithm

Complexity:

- The number of iterations to get accuracy ϵ is

$$O\left(\frac{n\sqrt{\log n}}{\epsilon}\right).$$

- At each iteration, the cost of computing a matrix exponential up to machine precision is $O(n^3)$.

Computing matrix exponentials:

- Many options, cf. "Nineteen Dubious Ways to Compute the Exponential of a Matrix" by Moler & Van Loan (2003).
- Padé approximation, full eigenvalue decomposition: $O(n^3)$ up to machine precision.
- In practice, machine precision is unnecessary. . .

First order algorithm

In d'Aspremont (2005): When minimizing a function with Lipschitz-continuous gradient using the method in Nesterov (1983), an **approximate gradient** is sufficient to get the $O(1/\epsilon)$ convergence rate. If the function and gradient approximations satisfy:

$$|f(x) - \tilde{f}(x)| \leq \delta \quad \text{and} \quad |\langle \tilde{\nabla} f(x) - \nabla f(x), y \rangle| \leq \delta \quad x, y \in Q_1,$$

we have:

$$f(x_k) - f(x^\star) \leq \frac{L d(x^\star)}{(k+1)(k+2)\sigma} + 10\delta$$

where L, $d(x^\star)$ and σ are problem constants.

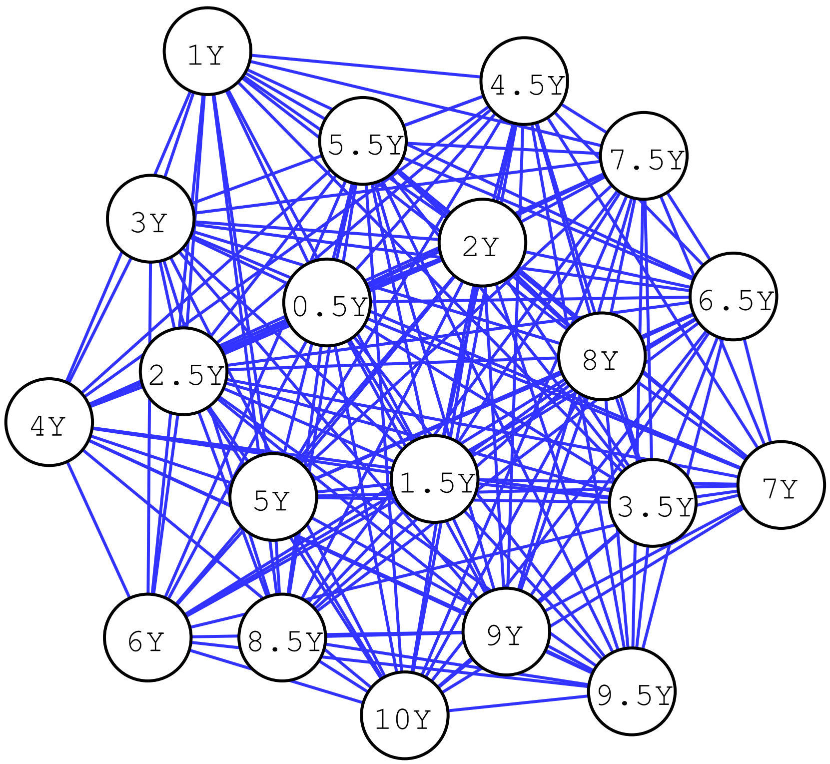
- Only a few dominant eigs. are required to get the matrix exponential.
- Dominant eigenvalues with ARPACK: **cubic** convergence.
- Optimal complexity of $O(1/\epsilon)$, same cost per iteration as regular methods with complexity $O(1/\epsilon^2)$.
- ARPACK exploits **sparsity**.

Outline

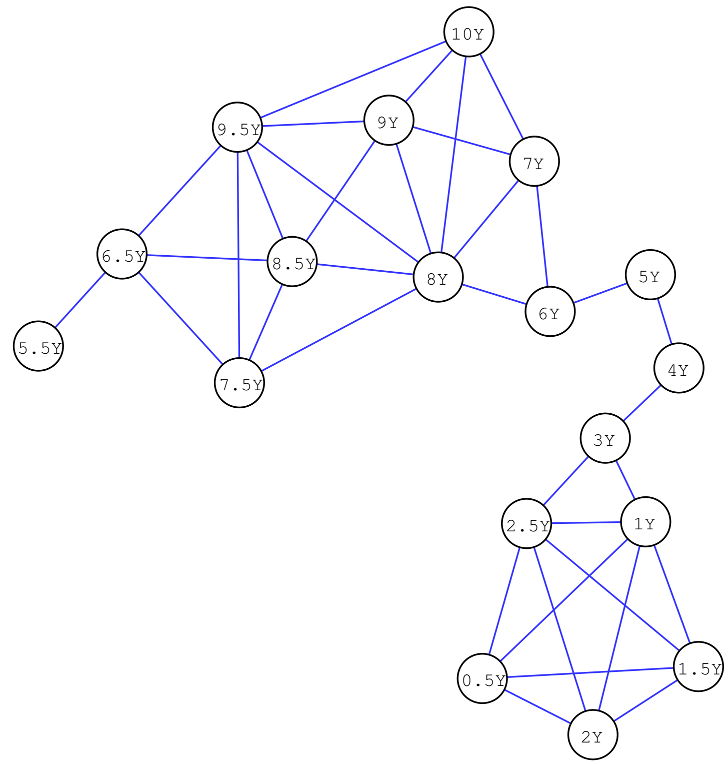
- Two classic relaxation tricks
    - Semidefinite relaxations and the lifting technique
    - The l_1 heuristic
- Applications
    - Covariance selection
    - Sparse PCA, SVD
    - Sparse nonnegative matrix factorization
- Solving large-scale semidefinite programs
    - First-order methods
    - **Numerical performance**

Covariance Selection

Forward rates covariance matrix for maturities ranging from 0.5 to 10 years.

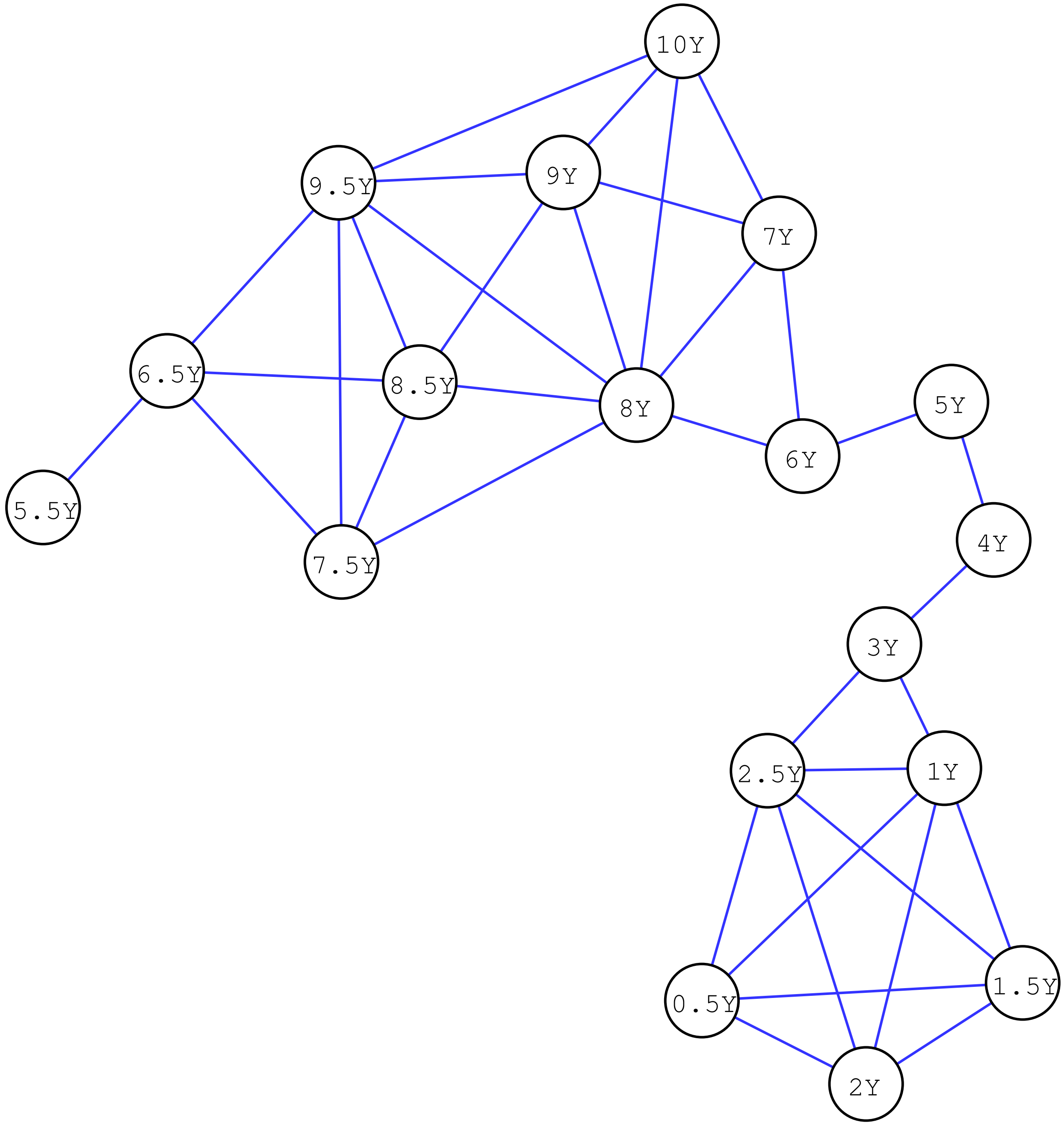


$\rho = 0$

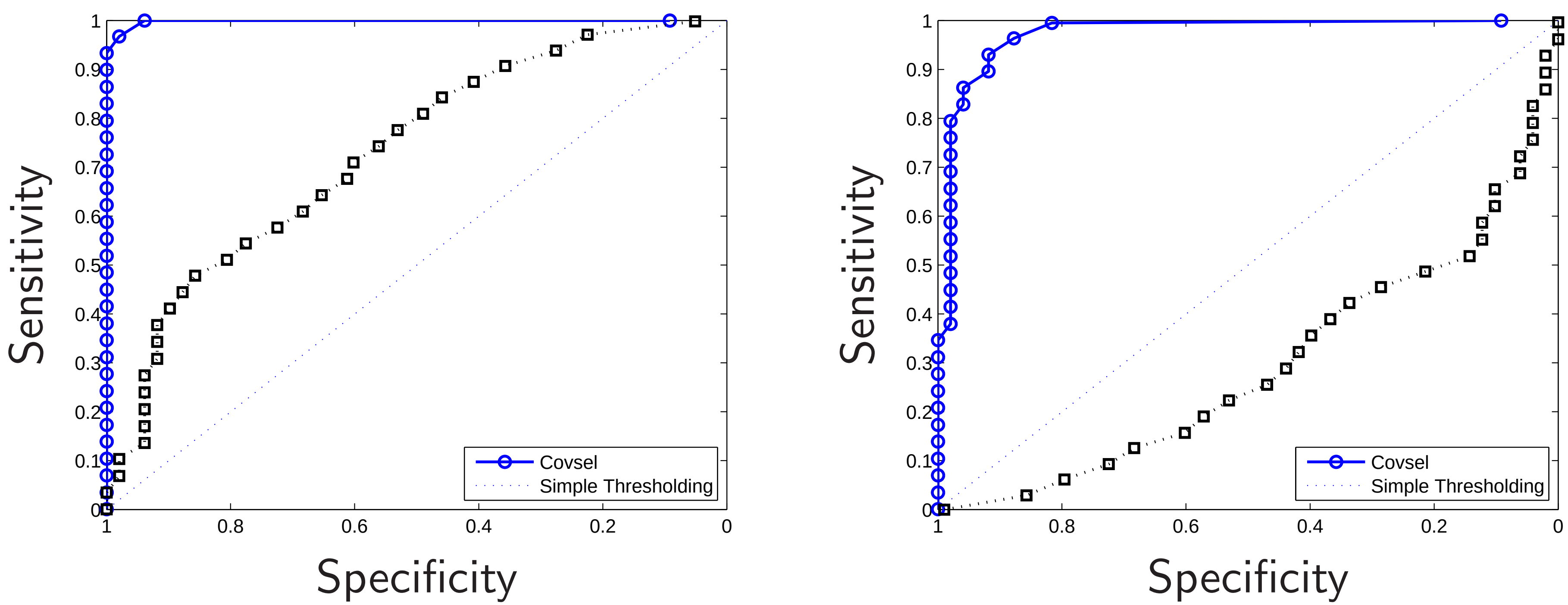


$\rho = .01$

Zoom. . .

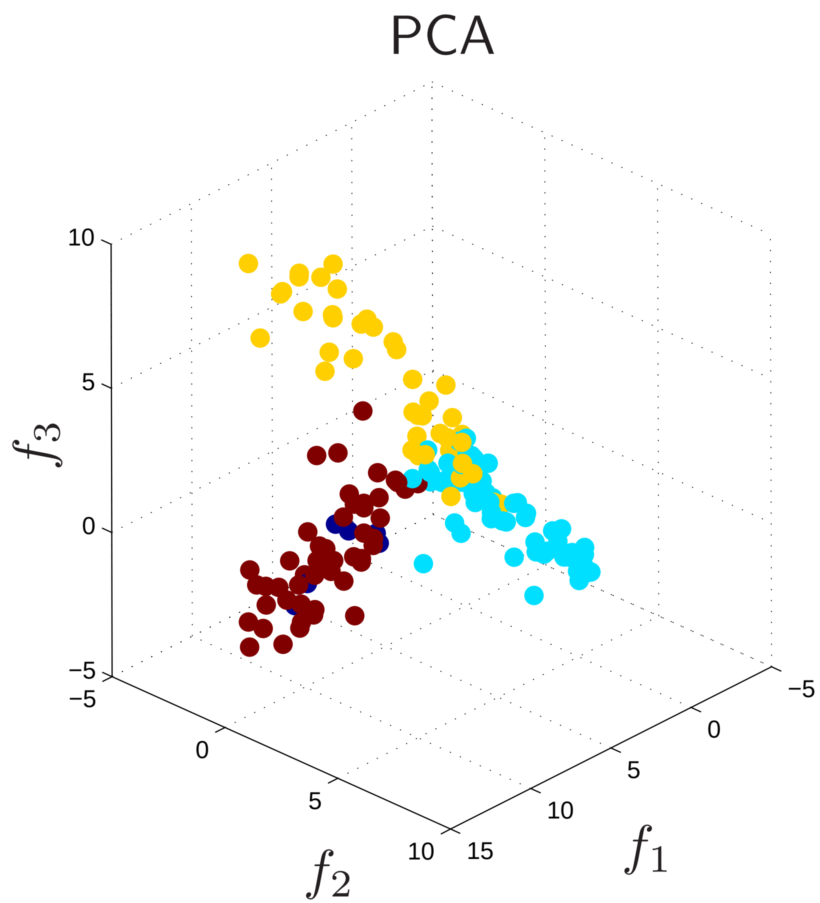


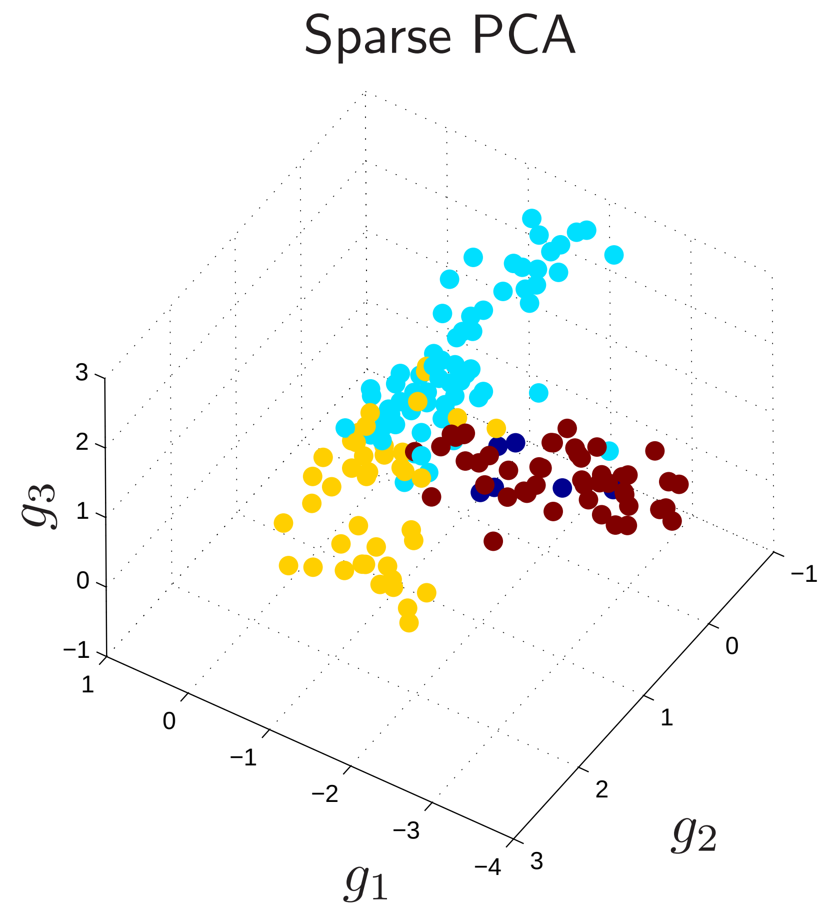
Covariance Selection



Classification Error. Sensitivity/Specificity curves for the solution to the covariance selection problem compared with a simple thresholding of B^{-1}, for various levels of noise: $\sigma = 0.3$ (left) and $\sigma = 0.5$ (right). Here $n = 50$.

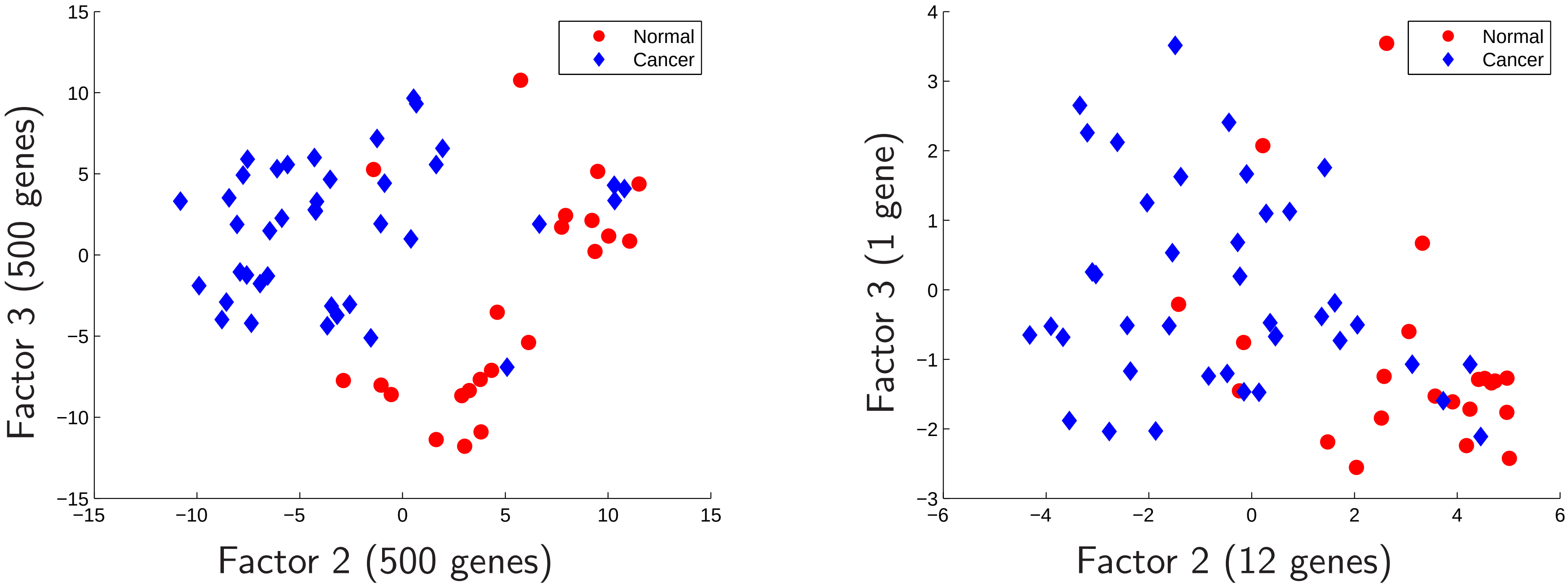
Sparse PCA





Clustering of the gene expression data in the PCA versus sparse PCA basis with 500 genes. The factors f on the left are dense and each use all 500 genes while the sparse factors $g_1,\ g_2$ and g_3 on the right involve 6, 4 and 4 genes respectively. (Data: Iconix Pharmaceuticals)

Sparse PCA



PCA Clustering (left) & DSPCA Clustering (right), colon cancer data set in Alon, Barkai, Notterman, Gish, Ybarra, Mack & Levine (1999).

Smooth first-order vs IP

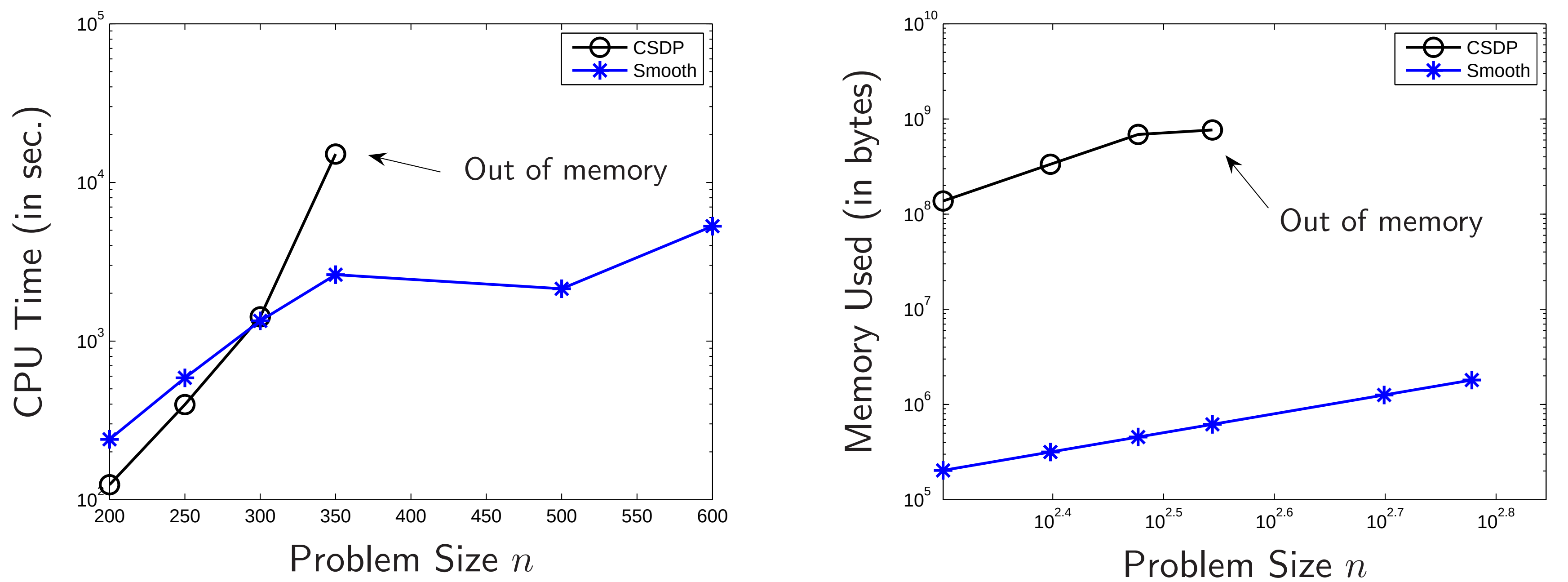
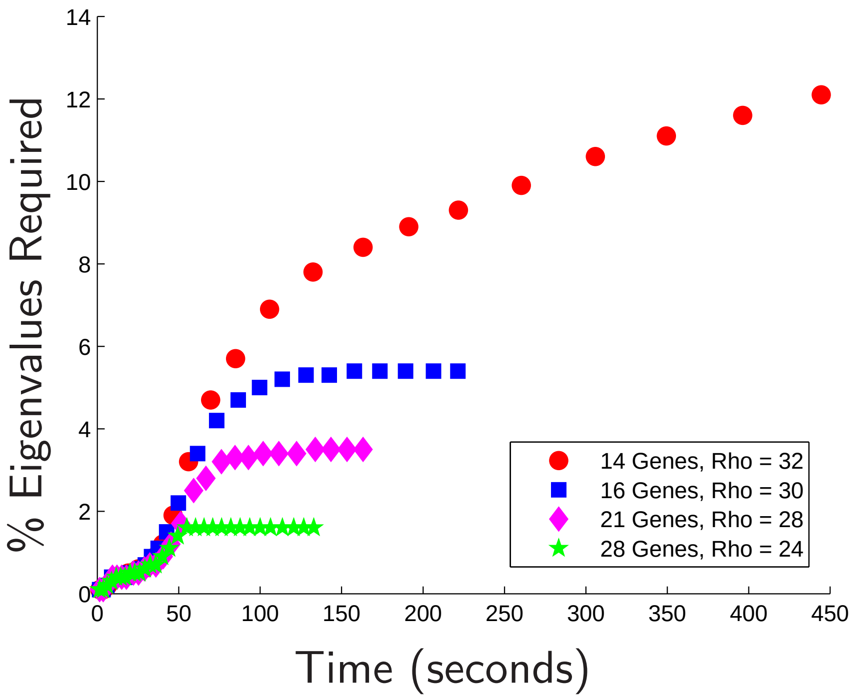
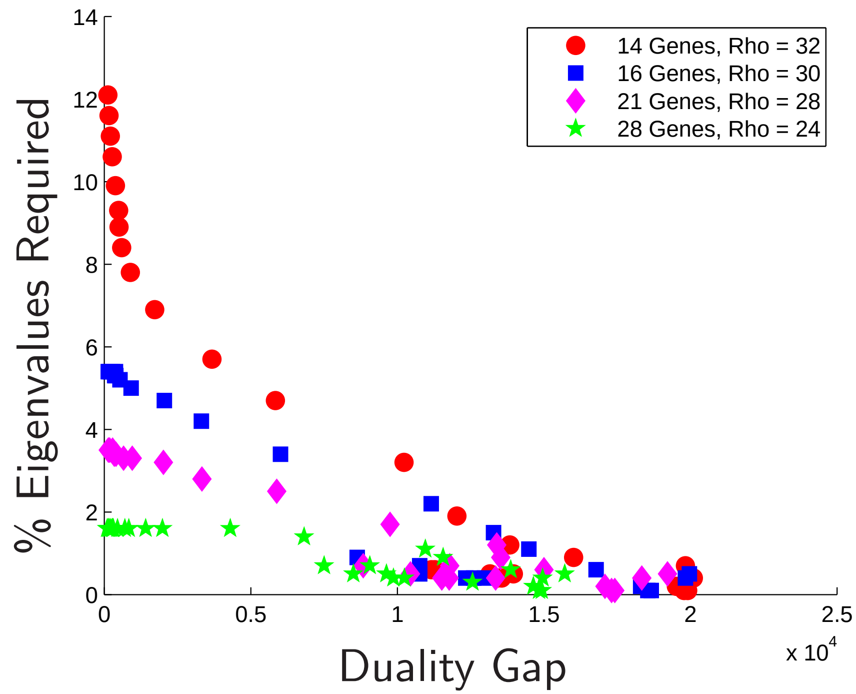


Figure 1: CPU time and memory usage versus n.

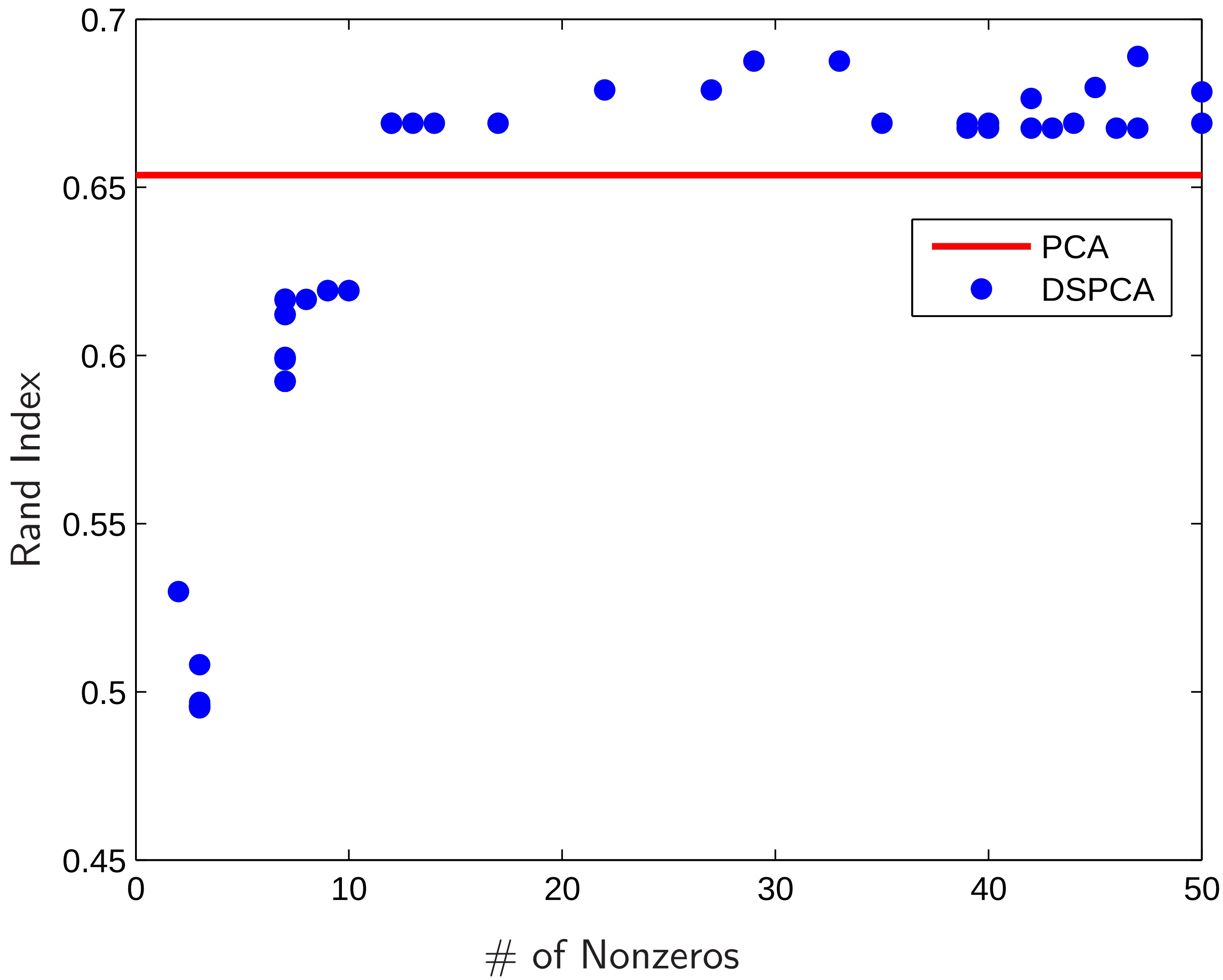
Sparse PCA





Eigenvalues vs. CPU Time (left), Duality Gap vs Eigs. (right), on 1000 genes.

Sparse PCA



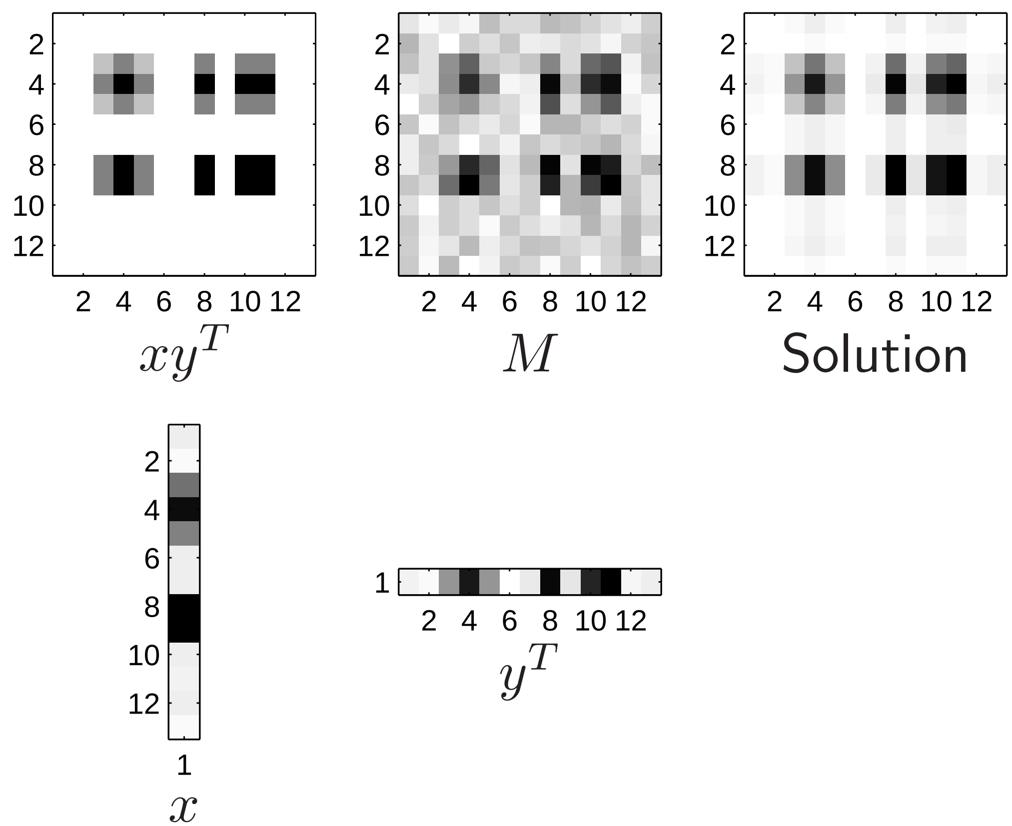
Sparsity versus Rand Index on colon cancer data set.

Sparse Nonnegative Matrix Factorization

Test relaxation on a matrix of the form:

$$M = xy^T + U$$

where U is uniform noise.



Conclusion

- Semidefinite relaxations of combinatorial problems in multivariate statistics.
- Infer sparse structural information on large datasets.
- Efficient codes can solve problems of with 10^3 variables in a few minutes.

Source code and binaries for sparse PCA (**DSPCA**) and covariance selection (**COVSEL**) available at:

`www.princeton.edu/~aspremon`

These slides are available at:

`www.princeton.edu/~aspremon/Banff07.pdf`

References

Akaike, J. (1973), Information theory and an extension of the maximum likelihood principle, *in* B. N. Petrov & F. Csaki, eds, 'Second international symposium on information theory', Akedemiai Kiado, Budapest, pp. 267–281.

Alon, A., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D. & Levine, A. J. (1999), 'Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays', *Cell Biology* **96**, 6745–6750.

Ben-Tal, A. & Nemirovski, A. (2001), *Lectures on modern convex optimization : analysis, algorithms, and engineering applications*, MPS-SIAM series on optimization, Society for Industrial and Applied Mathematics : Mathematical Programming Society, Philadelphia, PA.

Bilmes, J. A. (1999), 'Natural statistic models for automatic speech recognition', *Ph.D. thesis, UC Berkeley, Dept. of EECS, CS Division* .

Bilmes, J. A. (2000), 'Factored sparse inverse covariance matrices', *IEEE International Conference on Acoustics, Speech, and Signal Processing* .

Cadima, J. & Jolliffe, I. T. (1995), 'Loadings and correlations in the interpretation of principal components', *Journal of Applied Statistics* **22**, 203–214.

Candès, E. & Tao, T. (2005), 'Decoding by linear programming', *ArXiv: math.MG/0502327* .

Chen, S. S. & Gopinath, R. A. (1999), 'Model selection in acoustic modeling', *EUROSPEECH* .

Dahl, J., Roychowdhury, V. & Vandenberghe, L. (2005), 'Maximum likelihood estimation of gaussian graphical models: numerical implementation and topology selection', *UCLA preprint* .

d'Aspremont, A. (2005), 'Smooth optimization for sparse semidefinite programs', *ArXiv: math.OC/0512344* .

Dempster, A. (1972), 'Covariance selection', *Biometrics* **28**, 157–175.

Dobra, A., Hans, C., Jones, B., Nevins, J. J. R., Yao, G. & West, M. (2004), 'Sparse graphical models for exploring gene expression data', *Journal of Multivariate Analysis* **90**(1), 196–212.

Dobra, A. & West, M. (2004), 'Bayesian covariance selection', *working paper* .

Donoho, D. L. & Tanner, J. (2005), 'Sparse nonnegative solutions of underdetermined linear equations by linear programming', *Proceedings of the National Academy of Sciences* **102**(27), 9446–9451.

Fazel, M., Hindi, H. & Boyd, S. (2001), 'A rank minimization heuristic with application to minimum order system approximation', *Proceedings American Control Conference* **6**, 4734–4739.

Helmberg, C. & Rendl, F. (2000), 'A spectral bundle method for semidefinite programming', *SIAM Journal on Optimization* **10**(3), 673–696.

Jolliffe, I. T., Trendafilov, N. & Uddin, M. (2003), 'A modified principal component technique based on the LASSO', *Journal of Computational and Graphical Statistics* **12**, 531–547.

Lemaréchal, C. & Oustry, F. (1999), 'Semidefinite relaxations and Lagrangian duality with application to combinatorial optimization', *INRIA, Rapport de recherche* **3710**.

Lemaréchal, C. & Sagastizábal, C. (1997), 'Practical aspects of the Moreau-Yosida regularization: theoretical preliminaries', *SIAM Journal on Optimization* **7**(2), 367–385.

Moler, C. & Van Loan, C. (2003), 'Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later', *SIAM Review* **45**(1), 3–49.

Nemirovski, A. (2004), 'Prox-method with rate of convergence O(1/T) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems', *SIAM Journal on Optimization* **15**(1), 229–251.

Nesterov, Y. (1983), 'A method of solving a convex programming problem with convergence rate $O(1/k^2)$', *Soviet Mathematics Doklady* **27**(2), 372–376.

Nesterov, Y. (2005), 'Smooth minimization of nonsmooth functions', *Mathematical Programming, Series A* **103**, 127–152.

Nesterov, Y. & Nemirovskii, A. (1994), *Interior-point polynomial algorithms in convex programming*, Society for Industrial and Applied Mathematics, Philadelphia.

Sturm, J. (1999), 'Using SEDUMI 1.0x, a MATLAB toolbox for optimization over symmetric cones', *Optimization Methods and Software* **11**, 625–653.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the LASSO', *Journal of the Royal statistical society, series B* **58**(1), 267–288.

Toh, K. C., Todd, M. J. & Tutuncu, R. H. (1999), 'SDPT3 – a MATLAB software package for semidefinite programming', *Optimization Methods and Software* **11**, 545–581.

Zou, H., Hastie, T. & Tibshirani, R. (2004), 'Sparse principal component analysis', *To appear in Journal of Computational and Graphical Statistics* .