Short Course
Robust and Sparse Optimization
Part II: Sparse Optimization and Machine Learning Applications

Laurent El Ghaoui

EECS and IEOR Departments
UC Berkeley

PGMO Course, Fondation Mathématique Jacques Hadamard

Dec. 11, 2013

Outline

Outline

What is unsupervised learning?

In unsupervised learning, we are given a matrix of data points $X = [x_1, \ldots, x_m]$, with $x_i \in \mathbf{R}^n$; we wish to learn some condensed information from it.

Examples:

- Find one or several direction of maximal variance.
- Find a low-rank approximation or other structured approximation.
- Find correlations or some other statistical information (*e.g.*, graphical model).
- Find clusters of data points.

What is supervised learning?

In supervised learning, the data points are associated with “side” information that can “guide” (supervise) the learning process.

- In linear regression, each data point x_i is associated with a real number y_i (the “response”); the goal of learning is to fit the response vector to (say, linear) function of the data points, *e.g.* $y_i \approx w^T x_i$.
- In classification, the side information is a Boolean “label” (typically $y_i = \pm 1$); the goal is to find a set of coefficients such that the sign of a linear function $w^T x_i$ matches the values y_i.
- In structured output models, the side information is a more complex structure, such a tree.

Popular loss functions

- Squared loss: (for linear least-squares regression)

$$L(z, y) = \|z - y\|_2^2.$$

- Hinge loss: (for SVMs)

$$L(z, y) = \sum_{i=1}^{m} \max(0, 1 - y_i z_i)$$

- Logistic loss: (for logistic regression)

$$L(z, y) = -\sum_{i=1}^{m} \log(1 + e^{-y_i z_i}).$$

Outline

Generic sparse learning problem

Optimization problem with cardinality penalty:

$$\min_{w} \; L(X^T w) + \lambda \|w\|_0.$$

- Data: $X \in \mathbf{R}^{n \times m}$.
- Loss function L is convex.
- Cardinality function $\|w\|_0 := |\{j \; : \; w_j \neq 0\}|$ is non-convex.
- λ is a penalty parameter allowing to control sparsity.

- Arises in many applications, including (but not limited to) machine learning.
- Computationally intractable.

Classical approach

A now classical approach is to replace the cardinality function with an l_1-norm:

$$\min_w \; L(X^T w) + \lambda \|w\|_1 .$$

Pros:

- Problem becomes convex, tractable.
- Often works very well in practice.
- Many “recovery” results available.

Cons: may not work!

Recovery
A special case

Consider the sparse learning problem

$$\min_x \|w\|_0 \ : \ X^T w = y.$$

Assume optimal point is unique, let $w^{(0)}$ be the optimal point.

Now solve l_1-norm approximation

$$w^{(1)} := \arg\min_x \|w\|_1 \ : \ X^T w = y.$$

Since $w^{(1)}$ is feasible, we have $X^T(w^{(1)} - w^{(0)}) = 0$.

Facts: (see [2])

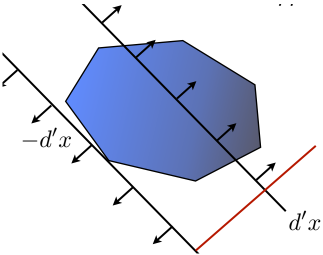
- Set of directions that decrease the norm from $w^{(1)}$ form a cone.
- If the nullspace of X^T does not intersect the cone, then $w^{(1)} = w^{(0)}$.

Mean width

Let $S \subseteq \mathbf{R}^n$ be a convex set, with support function

$$S_C(d) = \sup_{x \in S} \ d^T x.$$

Then $S_C(d) + S_C(-d)$ measures "width along direction d".



Mean width: with $\mathcal{S}^{n-1}$ be the unit Euclidean ball in $\mathbf{R}^n$,

$$\omega(C) := \mathbf{E}_u \ S_C(u) = \int_{u \in \mathcal{S}^{n-1}} S_C(u) du.$$

Gordon's escape theorem

When does a random subspace $\mathcal{A} \in \mathbf{R}^n$ intersect a convex cone C only at the origin?

Theorem: (Gordon, 1988) If

$$\mathbf{codim}(\mathcal{A}) \geq n \cdot \omega(C \cap \mathcal{S}^{n-1})^2,$$

then with high probability: $\mathcal{A} \cap C = \{0\}$.

Bounding mean width

A duality approach

$$
\begin{aligned}
\omega(C \cap \mathcal{S}^{n-1}) &= \mathbf{E}_u \max_{x \in C,\ \|x\|=1} u^T x \\
&\leq \mathbf{E}_u \max_{x \in C,\ \|x\| \leq 1} u^T x \\
&= \mathbf{E}_u \min_{v \in C^*} \|u - v\|,
\end{aligned}
$$

where C^* is the polar cone:

$$
C^* := \left\{ v \ : \ v^T u \leq 0 \text{ for every } u \in C \right\}.
$$

Name of the game is to *choose* an appropriate v.

Recovery rates

Fact: ([2]) Assume that the solution to cardinality problem with n variables and m constraints:

$$w^{(0)} = \arg\min_x \|w\|_0 \ : \ X^T w = y$$

is unique and has sparsity s. Using the l_1-norm approximation

$$w^{(1)} = \arg\min_x \|w\|_1 \ : \ X^T w = y,$$

the condition

$$m \geq 2s \log \frac{n}{s} + \frac{5}{4} s$$

guarantees that with high probability, $w^{(1)} = w^{(0)}$.

Similar results hold for a variety of norms (not just l_1).

Basic idea

LASSO and its dual

"Square-root" LASSO:

$$\min_w \|X^T w - y\|_2 + \lambda \|w\|_1.$$

with $X^T = [a_1, \ldots, a_n] \in \mathbf{R}^{m \times n}$, $y \in \mathbf{R}^m$, and $\lambda > 0$ are given. (Each $a_i \in \mathbf{R}^m$ corresponds to a variable in w, *i.e.* a "feature".)

Dual:

$$\max_\theta \ \theta^T y \ : \ \|\theta\|_2 \leq 1, \ \ |a_i^T \theta| \leq \lambda, \ \ i = 1, \ldots, n.$$

From optimality conditions, if at optimum in the dual the i-constraint is not active:

$$|a_i^T \theta| < \lambda$$

then $w_i = 0$ at optimum in the primal.

Basic idea

Safe Feature Elimination (SAFE)

From optimality:

$$|a_i^T \theta| < \lambda \Longrightarrow w_i = 0.$$

Since the dual problem involves the constraint $\|\theta\|_2 \leq 1$, the condition

$$\forall\, \theta, \quad \|\theta\|_2 \leq 1 \ : \ |a_i^T \theta| < \lambda$$

ensures that $w_i = 0$ at optimum.

SAFE condition:

$$\|a_i\|_2 < \lambda \Longrightarrow w_i = 0.$$

Advanced SAFE tests

Test can be strenghtened:

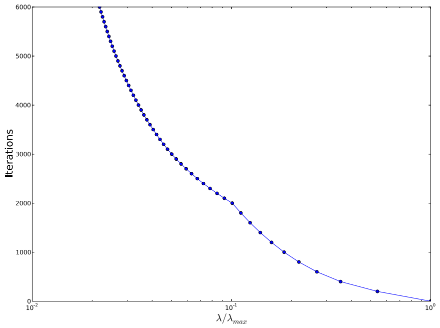
- Exploit optimal solution to problem for a higher value of λ.
- Use idea within the loop of a coordinate-descent (CD) algorithm.
- Allows to eliminate variables on the go.

Test is cheap:

- SAFE test costs as much as one iteration of gradient or CD method.
- Typically involves matrix-vector multiply $X^T w$, with w a sparse vector.

Experiment

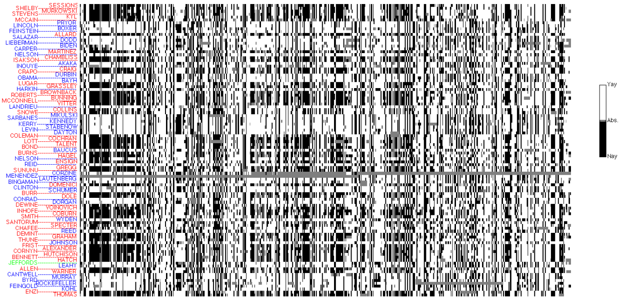
Data: KDD 2010b, 30M features, 20M documents. Target cardinality is 50.



- Applying SAFE in the loop of a coordinate-descent algorithm.
- Graph shows number of features involved to attain a given sparsity level.

Outline

Principal Component Analysis



Votes of US Senators, 2002-2004. The plot is impossible to read. . .

- Can we project data on a lower dimensional subspace?
- If so, how should we choose a projection?

Principal Component Analysis

Overview

Principal Component Analysis (PCA) originated in psychometrics in the 1930's. It is now widely used in

- Exploratory data analysis.
- Simulation.
- Visualization.

Application fields include

- Finance, marketing, economics.
- Biology, medecine.
- Engineering design, signal compression and image processing.
- Search engines, data mining.

Solution principles

PCA finds "principal components" (PCs), *i.e.* orthogonal directions of maximal variance.

- PCs are computed via EVD of covariance matrix.
- Can be interpreted as a "factor model" of original data matrix.

Variance maximization problem

Definition

Let us normalize the direction in a way that does not favor any direction.

Variance maximization problem:

$$\max_x \ \mathbf{var}(x) \ : \ \|x\|_2 = 1.$$

A non-convex problem!

Solution is easy to obtain via the eigenvalue decomposition (EVD) of S, or via the SVD of centered data matrix A_c.

Variance maximization problem

Solution

Variance maximization problem:

$$\max_x \ x^T S x \ : \ \|x\|_2 = 1.$$

Assume the EVD of S is given:

$$S = \sum_{i=1}^{p} \lambda_i u_i u_i^T,$$

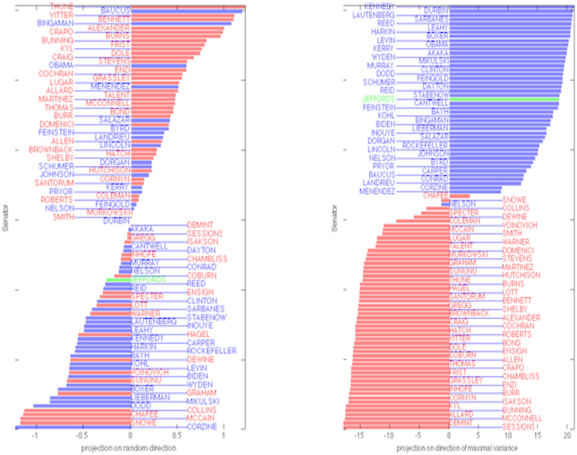
with $\lambda_1 \geq \ldots \lambda_p$, and $U = [u_1, \ldots, u_p]$ is orthogonal ($U^T U = I$). Then

$$\arg \max_{x \,:\, \|x\|_2 = 1} \ x^T S x = u_1,$$

where u_1 is any eigenvector of S that corresponds to the largest eigenvalue λ_1 of S.

Variance maximization problem

Example: US Senators voting data



Projection of US Senate voting data on random direction (left panel) and direction of maximal variance (right panel). The latter reveals party structure (party affiliations added after the fact). Note also the much higher range of values it provides.

Finding orthogonal directions
A deflation method

Once we've found a direction with high variance, can we repeat the process and find other ones?

Deflation method:

- Project data points on the subspace orthogonal to the direction we found.
- Fin a direction of maximal variance for projected data.

The process stops after p steps (p is the dimension of the whole space), but can be stopped earlier (to find only k directions, with $k << p$).

Finding orthogonal directions

Result

It turns out that the direction that solves

$$\max_x \ \mathbf{var}(x) \ : \ x^T u_1 = 0$$

is u_2, an eigenvector corresponding to the second-to-largest eigenvalue.

After k steps of the deflation process, the directions returned are $u_1, \ldots, u_k$.

Factor models

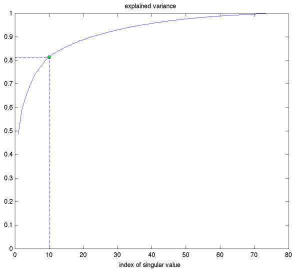
PCA allows to build a low-rank approximation to the data matrix:

$$A = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

Each v_i is a particular factor, and u_i's contain scalings.

Example

PCA of market data



Data: Daily log-returns of 77 Fortune 500 companies, 1/2/2007—12/31/2008.

- Plot shows the eigenvalues of covariance matrix in decreasing order.
- First ten components explain 80% of the variance.
- Largest magnitude of eigenvector for 1st component correspond to financial sector (FABC, FTU, MER, AIG, MS).

Sparse PCA: motivation

One of the issues with PCA is that it does not yield principal directions that are easily interpretable:

- The principal directions are really combinations of all the relevant features (say, assets).
- Hence we cannot interpret them easily.
- The previous thresholding approach (select features with large components, zero out the others) can lead to much degraded explained variance.

Sparse PCA

Problem definition

Modify the variance maximization problem:

$$\max_x \; x^T S x - \lambda \, \mathbf{Card}(x) \;:\; \|x\|_2 = 1,$$

where penalty parameter $\lambda \geq 0$ is given, and $\mathbf{Card}(x)$ is the cardinality (number of non-zero elements) in x.

The problem is hard but can be approximated via convex relaxation.

Safe feature elimination

Express S as $S = R^T R$, with $R = [r_1, \ldots, r_p]$ (each r_i corresponds to one feature).

Theorem (Safe feature elimination [3])

We have

$$\max_{x \,:\, \|x\|_2=1} x^T S x - \lambda \,\textbf{Card}(x) = \max_{z \,:\, \|z\|_2=1} \sum_{i=1}^{p} \max(0, (r_i^T z)^2 - \lambda).$$

SAFE

Corollary

If $\lambda > \|r_i\|_2^2 = S_{ii}$, we can safely remove the i-th feature (row/column of S).

- The presence of the penalty parameter allows to prune out dimensions in the problem.
- In practice, we want λ high as to allow better interpretability.
- Hence, interpretability requirement makes the problem easier in some sense!

Relaxation for sparse PCA

Step 1: l_1-norm bound

Sparse PCA problem:

$$\phi(\lambda) := \max_x \; x^T S x - \lambda \,\mathbf{Card}(x) \;\; : \;\; \|x\|_2 = 1,$$

First recall Cauchy-Schwartz inequality:

$$\|x\|_1 \leq \sqrt{\mathbf{Card}(x)}\|x\|_2,$$

hence we have the upper bound

$$\phi(\lambda) \leq \overline{\phi}(\lambda) := \max_x \; x^T S x - \lambda \|x\|_1^2 \;\; : \;\; \|x\|_2 = 1.$$

Relaxation for sparse PCA

Step 2: lifting and rank relaxation

Next we rewrite problem in terms of (PSD, rank-one) $X := xx^T$:

$$\overline{\phi} = \max_X \ \mathbf{Tr}\, SX - \lambda \|X\|_1 \ : \ X \succeq 0, \ \ \mathbf{Tr}\, X = 1, \ \ \mathbf{Rank}(X) = 1.$$

Drop the rank constraint , and get the upper bound

$$\overline{\lambda} \leq \psi(\lambda) := \max_X \ \mathbf{Tr}\, SX - \lambda \|X\|_1 \ : \ X \succeq 0, \ \ \mathbf{Tr}\, X = 1.$$

- Upper bound is a semidefinite program (SDP).
- In practice, X is found to be (close to) rank-one at optimum.

Sparse PCA Algorithms

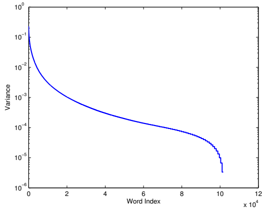
- The Sparse PCA problem remains challenging due to the huge number of variables.
- Second-order methods become quickly impractical as a result.
- SAFE technique often allows huge reduction in problem size.
- Dual block-coordinate methods are efficient in this case [9].
- Still area of active research. (Like SVD in the 70's-90's. . .)

Example 1

Sparse PCA of New York Times headlines

Data: NYTtimes text collection contains $300,000$ articles and has a dictionary of $102,660$ unique words.

The variance of the features (words) decreases very fast:



Sorted variances of 102,660 words in NYTimes data.

With a target number of words less than 10, SAFE allows to reduce the number of features from $n \approx 100,000$ to $n = 500$.

Example

Sparse PCA of New York Times headlines

Words associated with the top 5 sparse principal components in NYTimes

1st PC (6 words)	2nd PC (5 words)	3rd PC (5 words)	4th PC (4 words)	5th PC (4 words)
million	point	official	president	school
percent	play	government	campaign	program
business	team	united_states	bush	children
company	season	u_s	administration	student
market	game	attack		
companies				

Note: the algorithm found those terms without any information on the subject headings of the corresponding articles (unsupervised problem).

NYT Dataset

Comparison with thresholded PCA

Thresholded PCA involves simply thresholding the principal components.

$k = 2$	$k = 3$	$k = 9$	$k = 14$
even	even	even	would
like	like	we	new
	states	like	even
		now	we
		this	like
		will	now
		united	this
		states	will
		if	united
			states
			world
			so
			some
			if

1st PC from Thresholded PCA for various cardinality k. The results contain a lot of non-informative words.

Robust PCA

PCA is based on the assumption that the data matrix can be (approximately) written as a low-rank matrix:

$$A = LR^T,$$

with $L \in \mathbf{R}^{p \times k}$, $R \in \mathbf{R}^{m \times k}$, with $k << m, p$.

Robust PCA [1] assumes that A has a "low-rank plus sparse" structure:

$$A = N + LR^T$$

where "noise" matrix N is sparse (has many zero entries).

How do we discover N, L, R based on A?

Robust PCA model

In robust PCA, we solve the convex problem

$$\min_N \|A - N\|_* + \lambda \|N\|_1$$

where $\|\cdot\|_*$ is the so-called nuclear norm (sum of singular values) of its matrix argument. At optimum, $A - N$ has usually low-rank.

Motivation: the nuclear norm is akin to the l_1-norm of the vector of singular values, and l_1-norm minimization encourages sparsity of its argument.

CVX syntax

Here is a matlab snippet that solves a robust PCA problem via CVX, given integers n, m, a $n \times m$ matrix A and non-negative scalar λ exist in the workspace:

```
cvx_begin
variable X(n,m);
minimize( norm_nuc(A-X)+ lambda*norm(X(:),1))
cvx_end
```

Not the use of `norm_nuc`, which stands for the nuclear norm.

Outline

Motivation

We'd like to draw a graph that describes the links between the features (*e.g.*, words).

- Edges in the graph should exist when some strong, natural metric of similarity exist between features.
- For better interpretability, a *sparse* graph is desirable.
- Various motivations: portfolio optimization (with sparse risk term), clustering, etc.

Here we focus on exploring *conditional independence* within features.

Gaussian assumption

Let us assume that the data points are zero-mean, and follow a multi-variate Gaussian distribution: $x \simeq \mathcal{N}(0, \Sigma)$, with Σ a $p \times p$ covariance matrix. Assume Σ is positive definite.

Gaussian probability density function:

$$p(x) = \frac{1}{(2\pi \det \Sigma)^{p/2}} \exp((1/2)x^T \Sigma^{-1} x).$$

where $X := \Sigma^{-1}$ is the *precision* matrix.

Conditional independence

The pair of random variables x_i, x_j are *conditionally independent* if, for x_k fixed ($k \neq i, j$), the density can be factored:

$$p(x) = p_i(x_i) p_j(x_j)$$

where p_i, p_j depend also on the other variables.

- *Interpretation:* if all the other variables are fixed then x_i, x_j are independent.
- *Example:* Gray hair and shoe size are independent, conditioned on age.

Conditional independence

C.I. and the precision matrix

Theorem (C.I. for Gaussian RVs)

The variables x_i, x_j are conditionally independent if and only if the i, j element of the precision matrix is zero:

$$(\Sigma^{-1})_{ij} = 0.$$

Proof.

The coefficient of $x_i x_j$ in $\log p(x)$ is $(\Sigma^{-1})_{ij}$. ■

Sparse precision matrix estimation

Let us encourage sparsity of the precision matrix in the maximum-likelihood problem:

$$\max_X \ \log\det X - \mathbf{Tr}\, SX - \lambda\|X\|_1,$$

with $\|X\|_1 := \sum_{i,j} |X_{ij}|$, and $\lambda > 0$ a parameter.

- The above provides an invertible result, even if S is not positive-definite.
- The problem is convex, and can be solved in a large-scale setting by optimizing over column/rows alternatively.

Dual

Sparse precision matrix estimation:

$$\max_X \log\det X - \mathbf{Tr}\, SX - \lambda \|X\|_1.$$

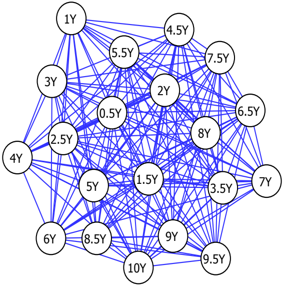
Dual:

$$\min_U -\log\det(S+U) \ : \ \|U\|_\infty \leq \lambda.$$

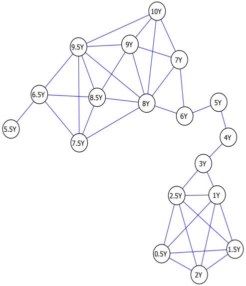
Block-coordinate descent: Minimize over one column/row of U cyclically. Each step is a QP.

Example

Data: Interest rates



Using covariance matrix ($\lambda = 0$).

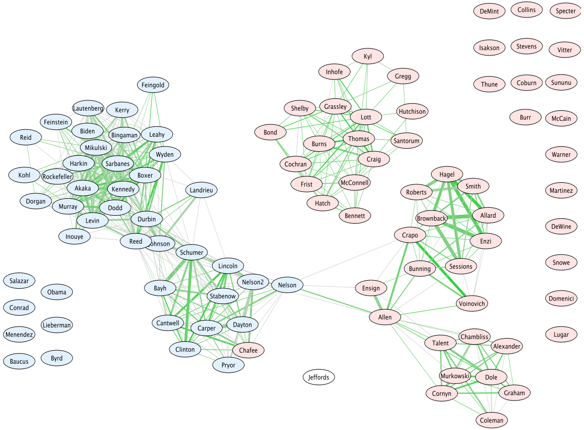


Using $\lambda = 0.1$.

The original precision matrix is dense, but the sparse version reveals the maturity structure.

Example

Data: US Senate voting, 2002-2004



Again the sparse version reveals information, here political blocks within each party.

Outline

Low-rank LP

Consider a linear programming problem in n variables with m constraints:

$$\min_x \; c^T x \; : \; Ax \leq b,$$

with $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$, and such that

- Many different problem instances involving the *same* matrix A have to be solved.
- The matrix A is close to low-rank.
- Clearly, we can approximate A with a low-rank matrix $A_{\rm lr}$ *once*, and exploit the low-rank structure to solve many instances of the LP fast.
- In doing so, we cannot guarantee that the solutions to the approximated LP are even feasible for the original problem.

Approach: robust low-rank LP

For the LP

$$\min_x \; c^T x \; : \; Ax \leq b,$$

with many instances of b, c:

- Invest in finding a low-rank approximation $A_{\rm lr}$ to the data matrix A, and estimate $\epsilon := \|A - A_{\rm lr}\|$.
- Solve the *robust counterpart*

$$\min_x \; c^T x \; : \; (A_{\rm lr} + \Delta)x \leq b \;\; \forall \, \Delta, \;\; \|\Delta\| \leq \epsilon.$$

- Robust counterpart can be written as SOCP

$$\min_{x,t} \; c^T x \; : \; A_{\rm lr} x + t\mathbf{1} \leq b, \;\; t \geq \|x\|_2.$$

- We can exploit the low-rank structure of $A_{\rm lr}$ and solve the above problem in time linear in $m + n$, for fixed rank.

Low-rank LASSO

In many learning problems, we need to solve many instances of the LASSO problem

$$\min_w \|X^T w - y\|_2 + \lambda \|w\|_1.$$

where

- For all the instances, the matrix X is a rank-one modification of the same matrix $\tilde{X}$.
- Matrix $\tilde{X}$ is close to low-rank (hence, X is).

In the topic imaging problem:

- $\tilde{X}$ is a term-by-document matrix that represents the whole corpus.
- y is one row of $\tilde{X}$ that encodes presence or absence of the topic in documents.
- X contains all remaining rows.

Robust low-rank LASSO

The robust low-rank LASSO

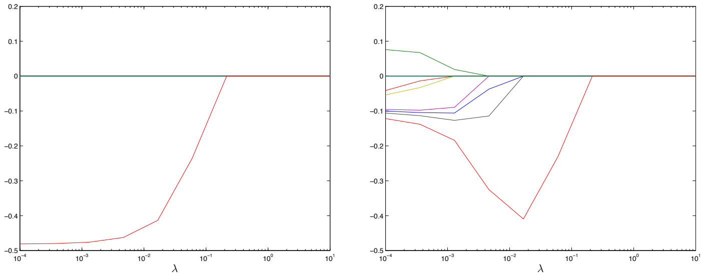
$$\min_{w} \max_{\|\Delta\| \leq \epsilon} \|(X_{\rm lr} + \Delta)^T w - y\|_2 + \lambda \|w\|_1$$

is expressed as a variant of "elastic net":

$$\min_{w} \|X_{\rm lr}^T w - y\|_2 + \lambda \|w\|_1 + \epsilon \|w\|_2.$$

- Solution can be found in time linear in $m + n$, for fixed rank.
- Solution has much better properties than low-rank LASSO, *e.g.* we can control the amount of sparsity.

Example



Rank-1 LASSO (left) and Robust Rank-1 LASSO (right) with random data. The plot shows the elements of the solution as a function of the l_1-norm penalty parameter.

- Without robustness ($\epsilon = 0$), the cardinality is 1 for $0 < \lambda < \lambda_{\max}$, where $\lambda_{\max}$ is a function of data. For $\lambda \geq \lambda_{\max}$, $w = 0$ at optimum. Hence the l_1-norm fails to control the solution.
- With robustness ($\epsilon = 0.01$), increasing λ allows to gracefully control the number of non-zeros in the solution.

Numerical experiments: low-rank approximation

Are real-world datasets approximately low-rank?

Dataset	TMC2007		RCV1V2		NYTIMES		PUBMED	
n		28,596		23,149		300,000		8,200,000
d		49,060		46,236		102,660		141,043
	Time (s)	σ_{k+1}/σ_1	Time (s)	σ_{k+1}/σ_1	Time (s)	σ_{k+1}/σ_1	Time (s)	σ_{k+1}/σ_1
k = 5	1	0.1539	1	0.2609	47	0.4095	187	0.4072
k = 10	1	0.1196	1	0.2100	50	0.3075	451	0.3494
k = 15	1	0.1010	1	0.1907	59	0.2709	520	0.3041
k = 20	2	0.0958	2	0.1769	73	0.2432	589	0.2793
k = 25	3	0.0909	3	0.1662	87	0.2312	687	0.2680
k = 30	4	0.0880	4	0.1615	93	0.2180	794	0.2580
k = 35	4	0.0858	4	0.1555	114	0.2098	932	0.2477
k = 40	5	0.0836	5	0.1507	130	0.2012	1150	0.2354
k = 45	6	0.0826	5	0.1475	142	0.1932	1208	0.2255
k = 50	7	0.0811	7	0.1430	158	0.1850	1862	0.2209

Runtimes[1] for computing a rank-k approximation to the whole data matrix.

[1] Experiments are conducted on a personal work station: 16GB RAM, 2.6GHz quad-core Intel.

Multi-label classification

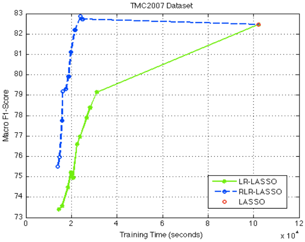
In multi-label classification, the task involves the same data matrix X, but many different response vectors y.

- Treat each label as a single classification subproblem (one-vs-all).
- Evaluation metric: Macro-F1 measure.
- Datasets:
  - RCV1-V2: 23,149 training documents; 781,265 test documents; 46,236 features; 101 labels.
  - TMC2007: 28,596 aviation safety reports; 49,060 features; 22 labels.

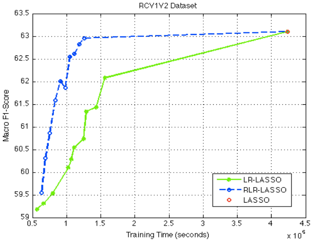
Multi-label classification

Plot performance vs. training times for various values of rank $k = 5, 10, \ldots, 50$.

TMC 2007 data set



RCV1V2 data set



In both cases, the low-rank robust counterpart allows to recover the performance obtained with full-rank LASSO (red dot), for a fraction of computing time.

Topic imaging

- Labels are columns of whole data matrix $\tilde{X}$.
- Compute low-rank approximation of $\tilde{X}$ when a column is removed.
- Evaluation: report predictive word lists for 10 queries.
- Datasets:
  - NYTimes: 300,000 documents; 102,660 features, file size is 1GB. Queries: 10 industry sectors.
  - PUBMED: 8,200,000 documents; 141,043 features, file size is 7.8GB. Queries: 10 diseases.
- In both cases we have pre-computed a rank k ($k = 20$) approximation using power iteration.

Topic imaging

automotive	agriculture	technology	tourism	aerospace	defence	financial	healthcare	petroleum	gaming
car	government	company	tourist	boeing	afghanistan	company	health	oil	game
vehicle	farm	computer	hotel	aircraft	attack	million	care	prices	gambling
auto	farmer	system	business	space	forces	stock	cost	gas	casino
sales	food	web	visitor	program	military	market	patient	fuel	player
model	water	information	economy	jet	gulf	money	corp	company	online
driver	trade	internet	travel	plane	troop	business	al_gore	barrel	computer
ford	land	american	tour	nasa	aircraft	firm	doctor	gasoline	tribe
driving	crop	job	local	flight	terrorist	fund	drug	bush	money
engine	economic	product	room	airbus	president	investment	medical	energy	playstation
consumer	country	software	plan	military	war	economy	insurance	opec	video

The New York Times data: Top 10 predictive words for different queries corresponding to industry sectors.

arthritis	asthma	cancer	depression	diabetes	gastritis	hiv	leukemia	migraines	parkinson
joint	bronchial	tumor	effect	diabetic	gastric	aid	cell	headache	treatment
synovial	asthmatic	treatment	treatment	insulin	h_pylori	infection	acute	headaches	effect
infection	children	carcinoma	disorder	level	chronic	cell	bone_marrow	pain	nerve
chronic	respiratory	cell	depressed	glucose	ulcer	hiv-1	leukemic	disorder	syndrome
pain	symptom	chemotherapy	pressure	control	acid	infected	tumor	women	disorder
treatment	allergic	survival	anxiety	plasma	stomach	antibodies	remission	chronic	neuron
fluid	infant	risk	symptom	diet	atrophic	risk	t_cell	duration	receptor
knee	inhalation	dna	drug	liver	antral	positive	antigen	symptom	alzheimer
acute	airway	malignant	neuron	renal	reflux	transmission	chemotherapy	gene	response
therapy	fev1	diagnosis	response	normal	treatment	drug	expression	therapy	brain

PubMed data: Top 10 predictive words for different queries corresponding to diseases.

Outline

StatNews project

At http://statnews.org/:

- Summarize large databases of news media.
- Predictive framework: relevant terms are found via sparse learning techniques.

Joint work with Bin Yu (Statistics, UC Berkeley).

Sparse predictive framework

To summarize how a topic is treated in a text database, we solve

$$\min_{w} \|X^T w - y\|_2 \ : \ \|w\|_0 \leq k$$

where

- X is $n \times m$ term-by-document matrix (containing *e.g.*, occurrences of terms across documents).
- m-vector y represents query term (*e.g.*, indicates occurrence of query term across all documents).
- w contains predictor coefficients; non-zeroes in w identify the few terms that are highly predictive of the query.
- $\|w\|_0$ stands for cardinality (number of non-zeros) of w; k is user-defined target cardinality (usually $k << m, n$, *e.g.*$k = 50$).

Staircase visualization

Image of Climate Change in Chinese Media

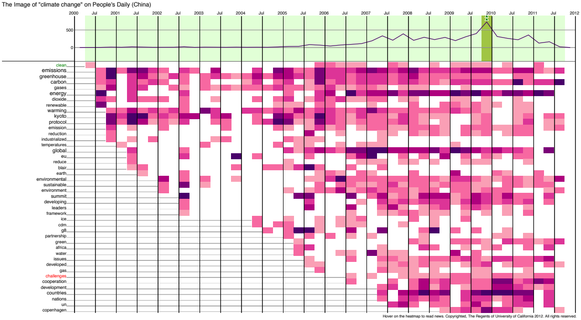


Image of topic "Climate change" over time. Each square encodes the size of regression coefficient in LASSO over a specific quarter. *Source:* People's Daily, 2000-2011.

Case study

Aviation safety reports

After each commercial flight in the US, pilots generate "ASRS reports" to document flight-related issues.

Key problem: detect emerging issues that are not being classified into existing categories, *e.g.*:

- "Wake vortex" problem of the Boeing 757.
- Increased number of runway incursions at LAX.

Case study

Aviation safety reports

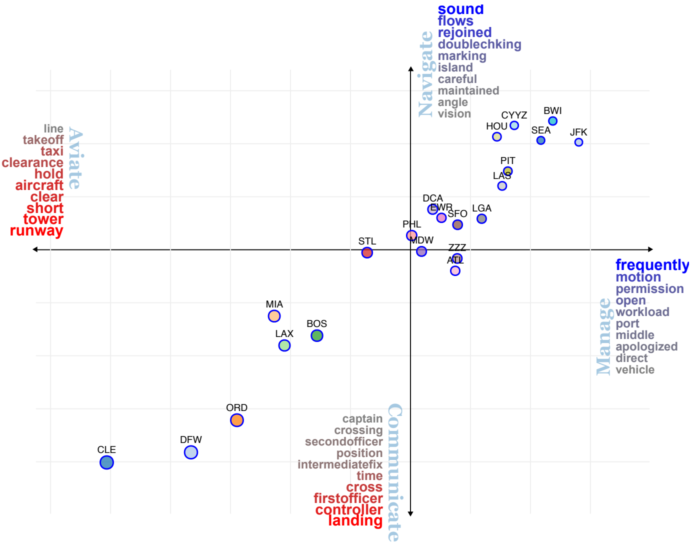
After each commercial flight in the US, pilots generate "ASRS reports" to document flight-related issues.

Key problem: detect emerging issues that are not being classified into existing categories, *e.g.*:

- "Wake vortex" problem of the Boeing 757.
- Increased number of runway incursions at LAX.

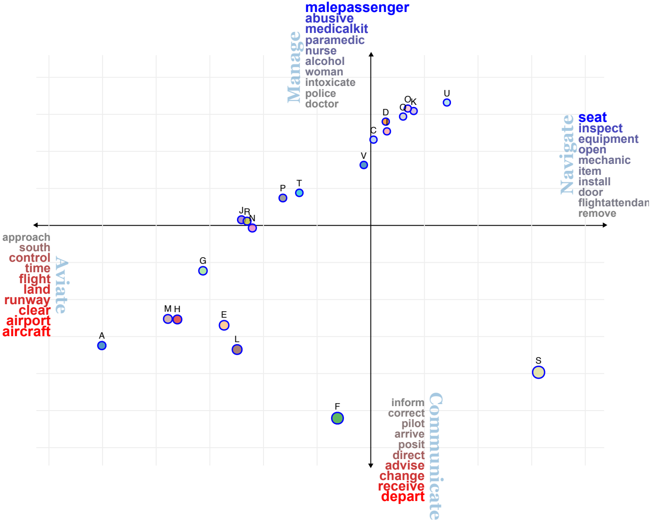
Don't search for a needle — picture the haystack!

Sparse PCA



Sparse PCA of runway-related reports in the ASRS database.

Recovering existing categories



Sparse PCA of ASRS reports, with categories shown.

Challenges and research topics

- Sparse learning:
  - Sparse supervised learning (LASSO) and unsupervised learning (sparse PCA).
  - Sparse probability optimization.
- Robust optimization:
  - Applications in data reduction.
  - Energy management.

Outline

Overview
Unsupervised learning
Supervised learning
Sparse supervised learning
Basics
Recovery
Safe Feature Elimination
Sparse PCA
Motivation
Example
SAFE
Relaxation
Algorithms
Examples
Variants
Sparse Covariance Selection
Sparse graphical models
Penalized maximum-likelihood
Example
Robust Optimization
Robust low-rank LP
Low-rank LASSO
StatNews
References

References I

[1] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright.
Robust principal component analysis?
2009.

[2] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan Willsky.
The convex geometry of linear inverse problems.
Foundations of Computational Mathematics, 12(6):805–849, 2012.

[3] Laurent El Ghaoui, Vivian Viallon, and Tarek Rabbani.
Safe feature elimination for the lasso and sparse supervised learning problems.
Pacific Journal of Optimization, (4):667–698, January 2012.
Special Issue on Conic Optimization.

[4] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre.
Generalized power method for sparse principal component analysis.
The Journal of Machine Learning Research, 11:517–553, 2010.

[5] Olivier Ledoit and Michael Wolf.
A well-conditioned estimator for large-dimensional covariance matrices.
Journal of Multivariate Analysis, 88:365–411, February 2004.

[6] O.Banerjee, L. El Ghaoui, and A. d'Aspremont.
Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data.
Journal of Machine Learning Research, 9:485–516, March 2008.

[7] S. Sra, S.J. Wright, and S. Nowozin.
Optimization for Machine Learning.
MIT Press, 2011.

[8] Y. Zhang, A. d'Aspremont, and L. El Ghaoui.
Sparse PCA: Convex relaxations, algorithms and applications.
In M. Anjos and J.B. Lasserre, editors, *Handbook on Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications*. Springer, 2011.
To appear.

[9] Y. Zhang and L. El Ghaoui.
Large-scale sparse principal component analysis and application to text data.
December 2011.