Optimization Models

EECS 127 / EECS 227AT

Laurent El Ghaoui

EECS department
UC Berkeley

Fall 2018

LECTURE 22b

Applications to Machine Learning (II): Unsupervised Learning

Learning:

Merriam–Webster Dictionary

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.

The empirical covariance matrix

Definition

Given a $p \times n$ data matrix $A = [a_1, \ldots, a_m]$ (each row representing say a log-return time-series over m time periods), the *empirical covariance matrix* is defined as the $p \times p$ matrix

$$S = \frac{1}{m}\sum_{i=1}^{m}(a_i - \hat{a})(a_i - \hat{a})^T, \;\; \hat{a} := \frac{1}{m}\sum_{i=1}^{m} a_i.$$

We can express S as

$$S = \frac{1}{m}A_c A_c^T,$$

where A_c is the *centered data matrix*, with p columns $(a_i - \hat{a})$, $i = 1, \ldots, m$.

The empirical covariance matrix

Link with directional variance

The (empirical) variance along direction x is

$$\mathrm{var}(x) = \frac{1}{m}\sum_{i=1}^{m}[x^T(a_i - \hat{a})]^2 = x^T S x = \frac{1}{m}\|A_c x\|_2^2.$$

where A_c is the centered data matrix.

Hence, covariance matrix gives information about variance along *any* direction.

Eigenvalue decomposition for symmetric matrices

Theorem 1 (EVD of symmetric matrices)

We can decompose any symmetric $p \times p$ matrix Q as

$$Q = \sum_{i=1}^{p} \lambda_i u_i u_i^T = U \Lambda U^T,$$

where $\Lambda = \mathrm{diag}\left(()\,\lambda_1, \ldots, \lambda_p\right)$, with $\lambda_1 \geq \ldots \geq \lambda_n$ the eigenvalues, and $U = [u_1, \ldots, u_p]$ is a $p \times p$ orthogonal matrix ($U^T U = I_p$) that contains the eigenvectors of Q. That is:

$$Q u_i = \lambda_i u_i, \quad i = 1, \ldots, p.$$

Singular Value Decomposition (SVD)

Theorem 2 (SVD of general matrices)

We can decompose any non-zero $p \times m$ matrix A as

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T = U \Sigma V^T, \;\; \Sigma = \mathrm{diag}\left(()\,\sigma_1, \ldots, \sigma_r, \underbrace{0, \ldots, 0}_{n\,-\,r\ \mathit{times}}\right)$$

where $\sigma_1 \geq \ldots \geq \sigma_r > 0$ are the singular values, and $U = [u_1, \ldots, u_m]$, $V = [v_1, \ldots, v_p]$ are square, orthogonal matrices ($U^T U = I_p$, $V^T V = I_m$). The first r columns of U, V contains the left- and right singular vectors of A, respectively, that is:

$$Av_i = \sigma_i u_i, \;\; A^T u_i = \sigma_i v_i, \;\; i = 1, \ldots, r.$$

Links between EVD and SVD

The SVD of a $p \times m$ matrix A is related to the EVD of a (PSD) matrix related to A.

If $A = U\Sigma V^T$ is the SVD of A, then

- The EVD of AA^T is $U\Lambda U^T$, with $\Lambda = \Sigma^2$.
- The EVD of A^TA is $V\Lambda V^T$.

Hence the left (resp. right) singular vectors of A are the eigenvectors of the PSD matrix AA^T (resp. A^TA).

Variational characterizations

Largest and smallest eigenvalues and singular values

If Q is square, symmetric:

$$\lambda_{\max}(Q) = \max_{x\,:\,\|x\|_2=1} x^T Q x.$$

If A is a general rectangular matrix:

$$\sigma_{\max}(A) = \max_{x\,:\,\|x\|_2=1} \|Ax\|_2.$$

Similar formulae for minimum eigenvalues and singular values.

Variational characterizations

Other eigenvalues and singular values

If Q is square, symmetric, the k-th largest eigenvalue satisfies

$$\lambda^k = \max_{x \in S^k, \,:\, \|x\|_2 = 1} x^T Q x,$$

where S^k is the subspace spanned by $\{u_k, \ldots, u_p\}$.

A similar result holds for singular values.

Low-rank approximation

For a given $p \times m$ matrix A, and integer $k \leq m, p$, the *k-rank approximation* problem is

$$A^{(k)} := \arg\min_X \ \|X - A\|_F \ : \ \mathbf{Rank}(X) \leq k,$$

where $\| \cdot \|_F$ is the Frobenius norm (Euclidean norm of the vector formed with all the entries of the matrix). The solution is

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

where $A = U \Sigma V^T$ is an SVD of the matrix A.

Low-rank approximation

Interpretation: rank-one case

Assume data matrix $A \in \mathbf{R}^{p \times m}$ represents time-series data (each row is a time-series). Assume also that A is rank-one, that is, $A = uv^T \in \mathbf{R}^{p \times m}$, where u, v are vectors. Then

$$A = \begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix}, \;\; a_j(t) = u(j)v(t), \;\; 1 \leq j \leq p, \;\; 1 \leq t \leq m.$$

Thus, each time-series is a "scaled" copy of the time-series represented by v, with scaling factors given in u. We can think of v as a "factor" that drives all the time-series.

Low-rank approximation

Interpretation: low-rank case

When A is rank k, that is,

$$A = UV^T, \quad U \in \mathbf{R}^{p \times k}, \quad V \in \mathbf{R}^{m \times k}, \quad k << m, p,$$

we can express the j-th row of A as

$$a_j(t) = \sum_{i=1}^{k} u_i(j) v_i(t), \quad 1 \leq j \leq p, \quad 1 \leq t \leq m.$$

Thus, each time-series is the sum of scaled copies of k time-series represented by $v_1, \ldots, v_k$, with scaling factors given in $u_1, \ldots, u_k$. We can think of v_i's as the few "factors" that drive all the time-series.

Motivation

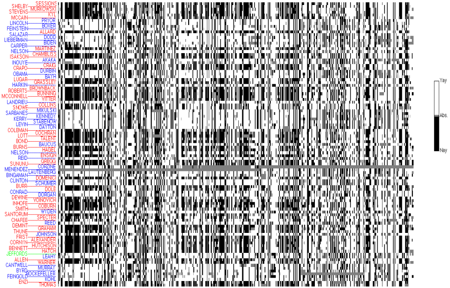


Figure: Votes of US Senators, 2002-2004. The plot is impossible to read. . .

- 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:

$$\max_{x} \ \mathrm{var}(x) \ : \ \|x\|_2 = 1.$$

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:

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

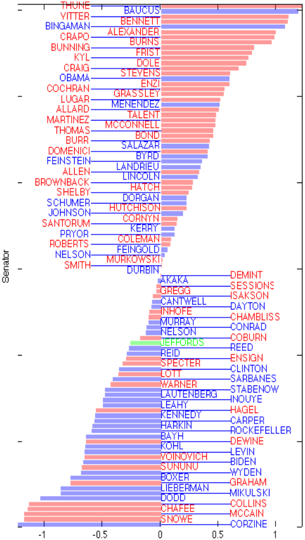
with $\lambda_1 \geq \ldots \lambda_p$, and $U = [u_1, \ldots, u_p]$ is orthogonal ($U^T U = I$). Then

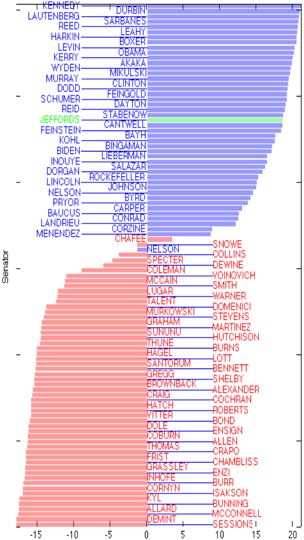
$$\arg \max_{x \, : \, \|x\|_2 = 1} \; x^T S x = u_1,$$

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





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

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

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

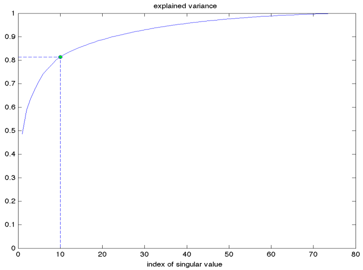


Figure: Data: Daily log-returns of 77 Fortune 500 companies, 1/2/2007—12/31/2008.

- 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).

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 **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 3 (Safe feature elimination [2])

We have

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

SAFE

Corollary 1

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 [7].
- 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:

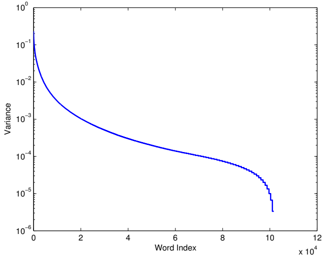


Figure: Sorted variances of 102,660 words in NYTimes data.

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

Table: Words associated with the top 5 sparse principal components in NYTimes

1st PC	2nd PC	3rd PC	4th PC	5th PC
(6 words)	(5 words)	(5 words)	(4 words)	(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

Table: 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.

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 4 (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:

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

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

Example

Data: Interest rates

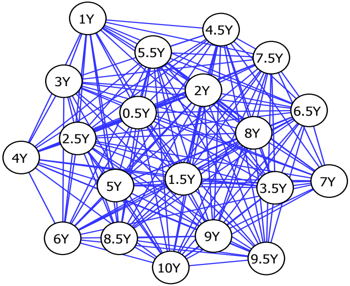


Figure: Using covariance matrix ($\lambda = 0$).

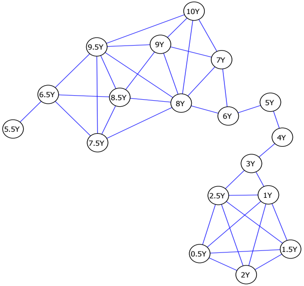
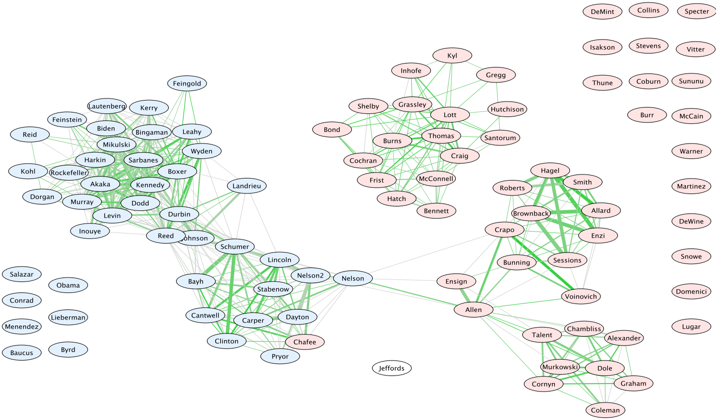


Figure: Using $\lambda = 0.1$.

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.

References

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

L. El Ghaoui.
On the quality of a semidefinite programming bound for sparse principal component analysis.
arXiv:math/060144, February 2006.

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

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.

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

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.

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