

PCA, Kernel PCA, ICA

Learning Representations.

Dimensionality Reduction.

Maria-Florina Balcan

04/08/2015

# Big & High-Dimensional Data

- High-Dimensions = Lot of Features

## Document classification

Features per document =  
thousands of words/unigrams  
millions of bigrams, contextual  
information



## Surveys - Netflix

480189 users x 17770 movies

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6
Tom	5	?	?	1	3	?
George	?	?	3	1	2	5
Susan	4	3	1	?	5	1
Beth	4	3	?	2	4	2

# Big & High-Dimensional Data

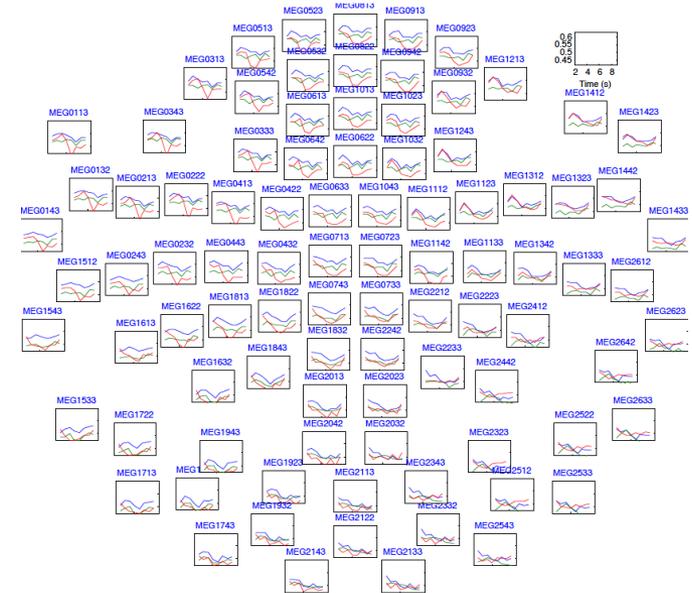
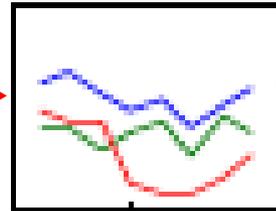
- High-Dimensions = Lot of Features

## MEG Brain Imaging

120 locations x 500 time points  
x 20 objects



MEG0633



Or any high-dimensional image data



- Big & High-Dimensional Data.
- Useful to learn lower dimensional representations of the data.

# Learning Representations

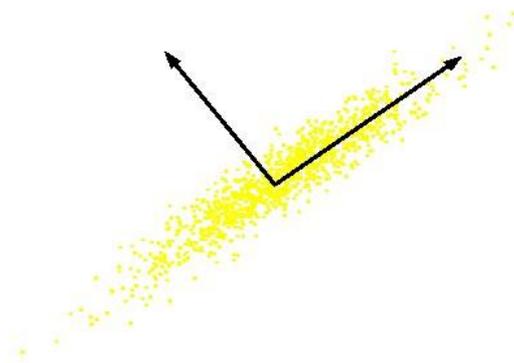
**PCA, Kernel PCA, ICA:** Powerful unsupervised learning techniques for extracting hidden (potentially lower dimensional) structure from high dimensional datasets.

## Useful for:

- Visualization
- More efficient use of resources (e.g., time, memory, communication)
- Statistical: fewer dimensions  $\rightarrow$  better generalization
- Noise removal (improving data quality)
- Further processing by machine learning algorithms

# Principal Component Analysis (PCA)

**What is PCA:** Unsupervised technique for extracting variance structure from high dimensional datasets.

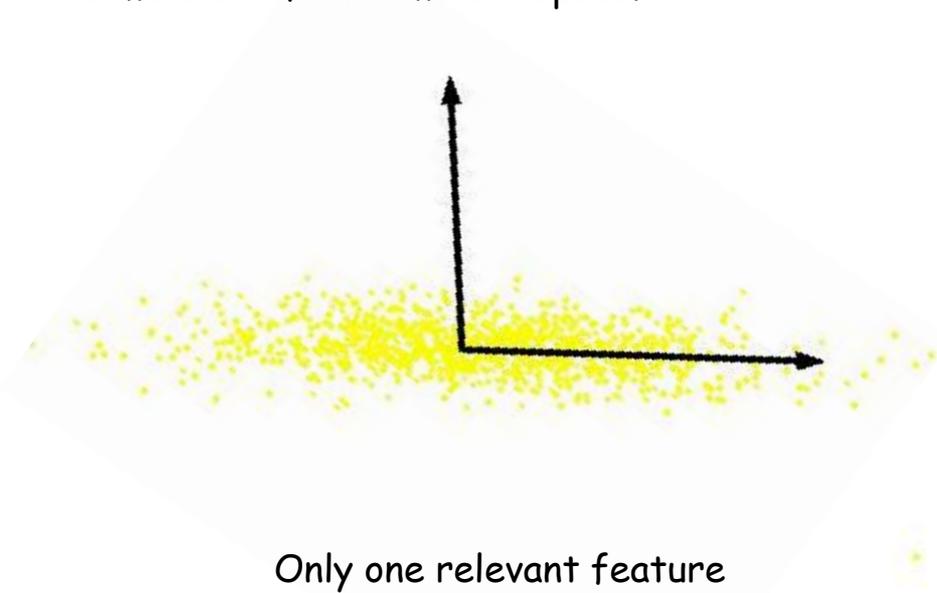


- PCA is an orthogonal projection or transformation of the data into a (possibly lower dimensional) subspace so that the variance of the projected data is maximized.

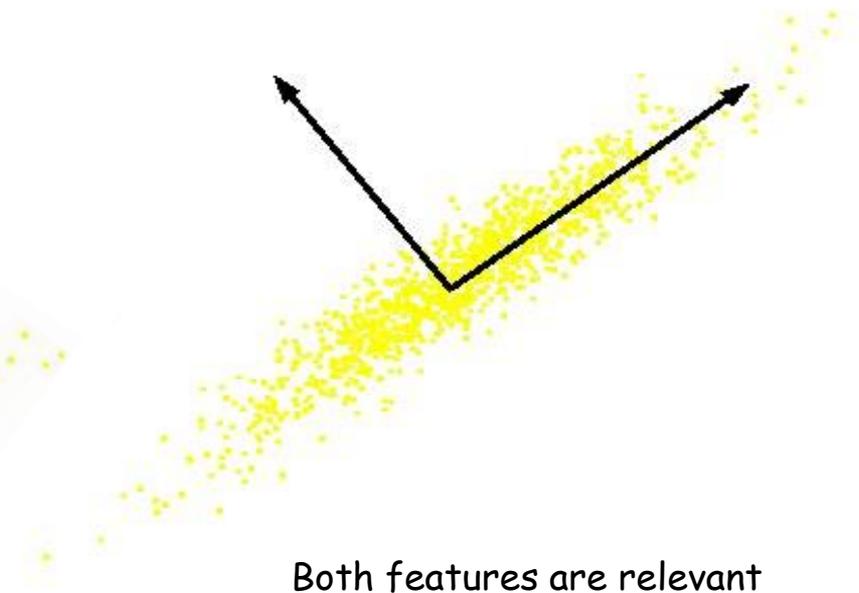
# Principal Component Analysis (PCA)

Intrinsically lower dimensional than the dimension of the ambient space.

If we rotate data, again only one coordinate is more important.



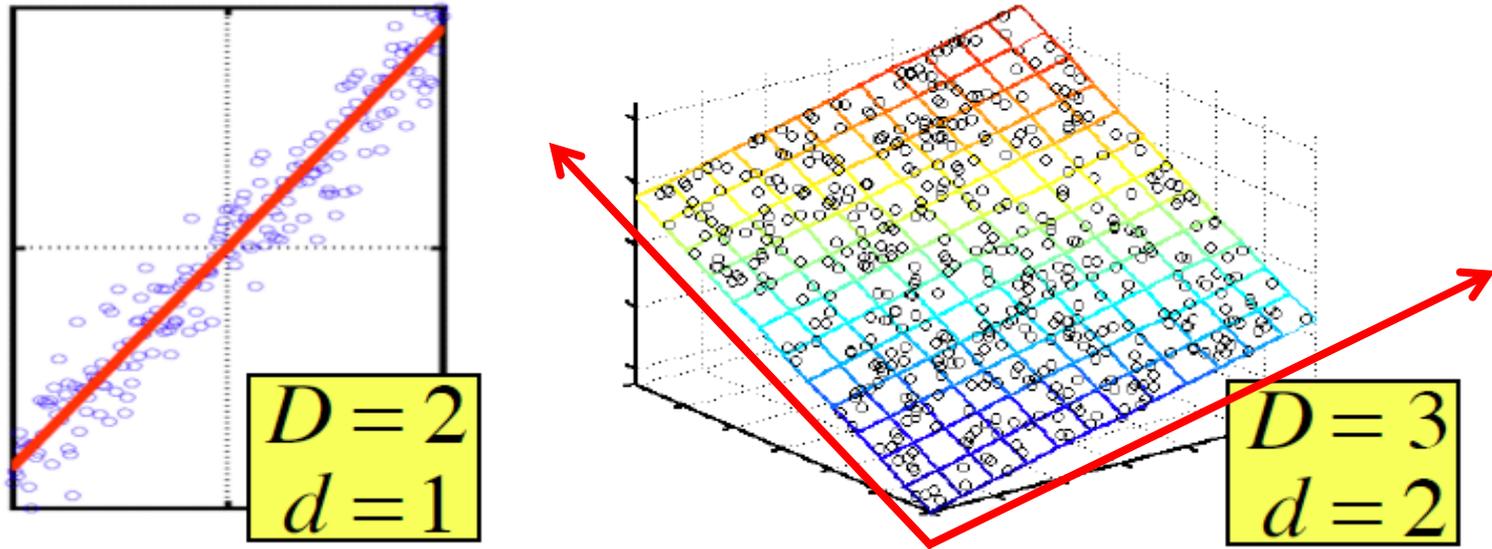
Only one relevant feature



Both features are relevant

Question: Can we transform the features so that we only need to preserve one latent feature?

# Principal Component Analysis (PCA)



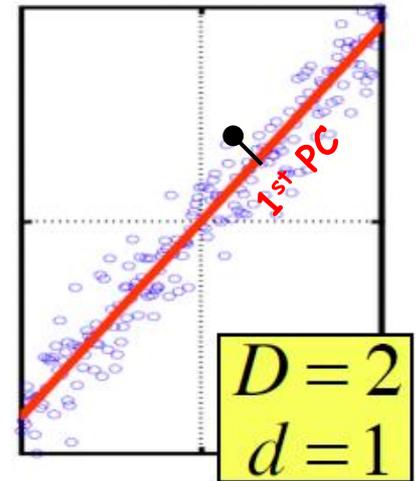
In case where data lies on or near a low  $d$ -dimensional linear subspace, axes of this subspace are an effective representation of the data.

Identifying the axes is known as **Principal Components Analysis**, and can be obtained by using classic matrix computation tools (Eigen or Singular Value Decomposition).

# Principal Component Analysis (PCA)

Principal Components (PC) are orthogonal directions that capture most of the variance in the data.

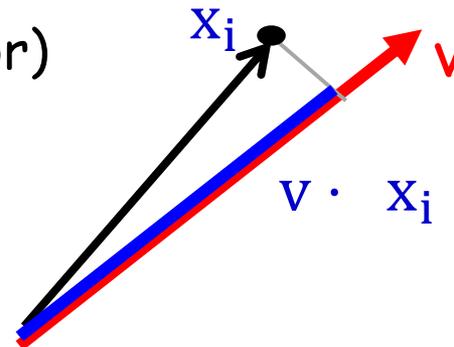
- First PC - direction of greatest variability in data.
- Projection of data points along first PC discriminates data most along any one direction (pts are the most spread out when we project the data on that direction compared to any other directions).



Quick reminder:

$\|v\|=1$ , Point  $x_i$  (D-dimensional vector)

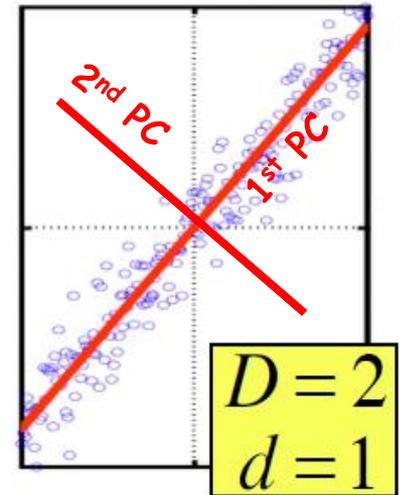
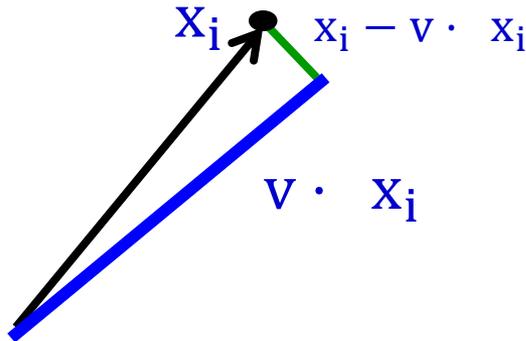
Projection of  $x_i$  onto  $v$  is  $v \cdot x_i$



# Principal Component Analysis (PCA)

Principal Components (PC) are orthogonal directions that capture most of the variance in the data.

- 1<sup>st</sup> PC - direction of greatest variability in data.



- 2<sup>nd</sup> PC - Next orthogonal (uncorrelated) direction of greatest variability

(remove all variability in first direction, then find next direction of greatest variability)

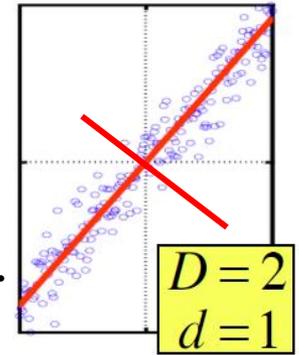
- And so on ...

# Principal Component Analysis (PCA)

Let  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d$  denote the  $d$  principal components.

$$\mathbf{v}_i \cdot \mathbf{v}_j = 0, i \neq j \quad \text{and} \quad \mathbf{v}_i \cdot \mathbf{v}_i = 1, i = j$$

Assume data is centered (we extracted the sample mean).



Let  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$  (columns are the datapoints)

Find vector that maximizes sample variance of projected data

$$\frac{1}{n} \sum_{i=1}^n (\mathbf{v}^T \mathbf{x}_i)^2 = \mathbf{v}^T \mathbf{X} \mathbf{X}^T \mathbf{v}$$

$$\max_{\mathbf{v}} \mathbf{v}^T \mathbf{X} \mathbf{X}^T \mathbf{v} \quad \text{s.t.} \quad \mathbf{v}^T \mathbf{v} = 1$$

$$\text{Lagrangian: } \max_{\mathbf{v}} \mathbf{v}^T \mathbf{X} \mathbf{X}^T \mathbf{v} - \lambda \mathbf{v}^T \mathbf{v}$$

Wrap constraints into the objective function

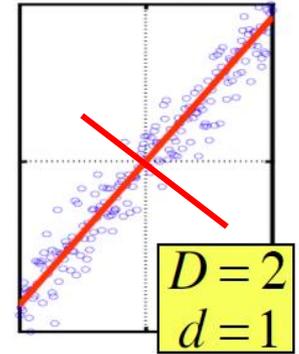
$$\partial / \partial \mathbf{v} = 0 \quad (\mathbf{X} \mathbf{X}^T - \lambda \mathbf{I}) \mathbf{v} = 0 \quad \Rightarrow \quad \boxed{(\mathbf{X} \mathbf{X}^T) \mathbf{v} = \lambda \mathbf{v}}$$

# Principal Component Analysis (PCA)

$(X X^T)v = \lambda v$ , so  $v$  (the first PC) is the eigenvector of sample correlation/covariance matrix  $X X^T$

Sample variance of projection  $v^T X X^T v = \lambda v^T v = \lambda$

Thus, the eigenvalue  $\lambda$  denotes the amount of variability captured along that dimension (aka amount of energy along that dimension).

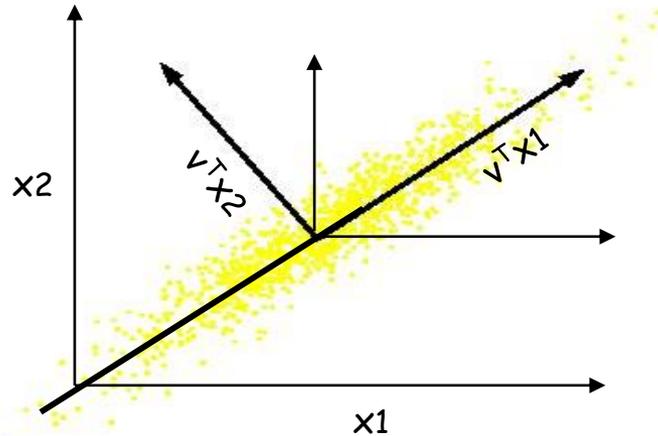


Eigenvalues  $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots$

- The 1<sup>st</sup> PC  $v_1$  is the the eigenvector of the sample covariance matrix  $X X^T$  associated with the largest eigenvalue
- The 2<sup>nd</sup> PC  $v_2$  is the the eigenvector of the sample covariance matrix  $X X^T$  associated with the second largest eigenvalue
- And so on ...

# Principal Component Analysis (PCA)

- So, the new axes are the eigenvectors of the matrix of sample correlations  $XX^T$  of the data.
- Transformed features are uncorrelated.



- Geometrically: centering followed by rotation.
  - Linear transformation

**Key computation:** eigendecomposition of  $XX^T$  (closely related to SVD of  $X$ ).

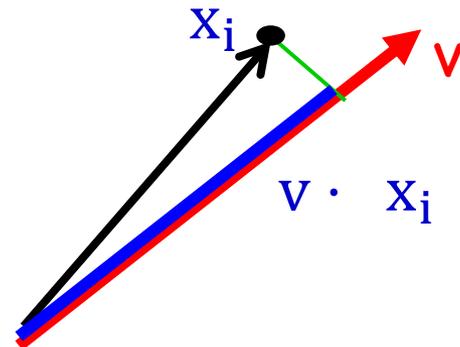
# Two Interpretations

So far: **Maximum Variance Subspace**. PCA finds vectors  $v$  such that projections on to the vectors capture maximum variance in the data

$$\frac{1}{n} \sum_{i=1}^n (\mathbf{v}^T \mathbf{x}_i)^2 = \mathbf{v}^T \mathbf{X} \mathbf{X}^T \mathbf{v}$$

Alternative viewpoint: **Minimum Reconstruction Error**. PCA finds vectors  $v$  such that projection on to the vectors yields minimum MSE reconstruction

$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - (\mathbf{v}^T \mathbf{x}_i) \mathbf{v}\|^2$$



# Two Interpretations

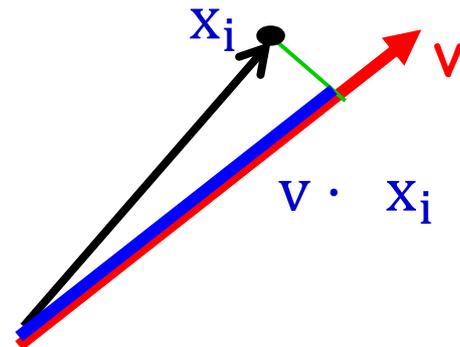
E.g., for the first component.

**Maximum Variance Direction:** 1<sup>st</sup> PC a vector  $v$  such that projection on to this vector capture maximum variance in the data (out of all possible one dimensional projections)

$$\frac{1}{n} \sum_{i=1}^n (\mathbf{v}^T \mathbf{x}_i)^2 = \mathbf{v}^T \mathbf{X} \mathbf{X}^T \mathbf{v}$$

**Minimum Reconstruction Error:** 1<sup>st</sup> PC a vector  $v$  such that projection on to this vector yields minimum MSE reconstruction

$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - (\mathbf{v}^T \mathbf{x}_i) \mathbf{v}\|^2$$



# Why? Pythagorean Theorem

E.g., for the first component.

**Maximum Variance Direction:** 1<sup>st</sup> PC a vector  $v$  such that projection on to this vector capture maximum variance in the data (out of all possible one dimensional projections)

$$\frac{1}{n} \sum_{i=1}^n (\mathbf{v}^T \mathbf{x}_i)^2 = \mathbf{v}^T \mathbf{X} \mathbf{X}^T \mathbf{v}$$

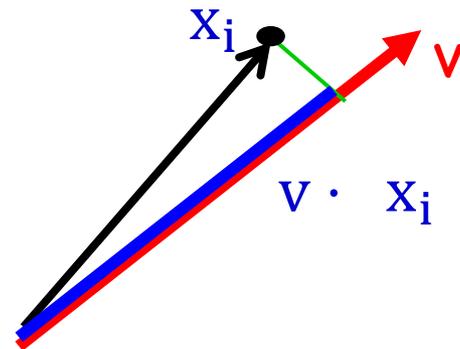
$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - (\mathbf{v}^T \mathbf{x}_i) \mathbf{v}\|^2$$

**Minimum Reconstruction Error:** 1<sup>st</sup> PC a vector  $v$  such that projection on to this vector yields minimum MSE reconstruction

$$\text{blue}^2 + \text{green}^2 = \text{black}^2$$

black<sup>2</sup> is fixed (it's just the data)

So, maximizing blue<sup>2</sup> is equivalent to minimizing green<sup>2</sup>

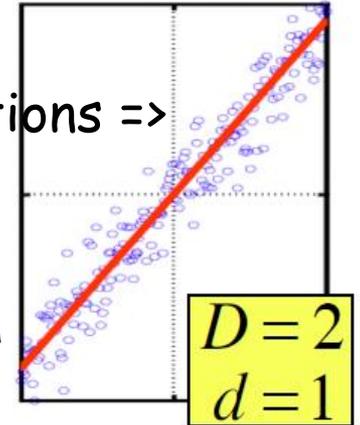


# Dimensionality Reduction using PCA

The eigenvalue  $\lambda$  denotes the amount of variability captured along that dimension (aka amount of energy along that dimension).

Zero eigenvalues indicate no variability along those directions => data lies exactly on a linear subspace

Only keep data projections onto principal components with non-zero eigenvalues, say  $v_1, \dots, v_k$ , where  $k = \text{rank}(X X^T)$



## Original representation

Data point

$$x_i = (x_i^1, \dots, x_i^D)$$

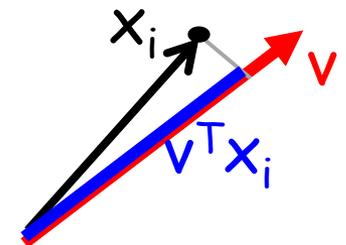
D-dimensional vector

## Transformed representation

projection

$$(v_1 \cdot x_i^1, \dots, v_d \cdot x_i^d)$$

d-dimensional vector

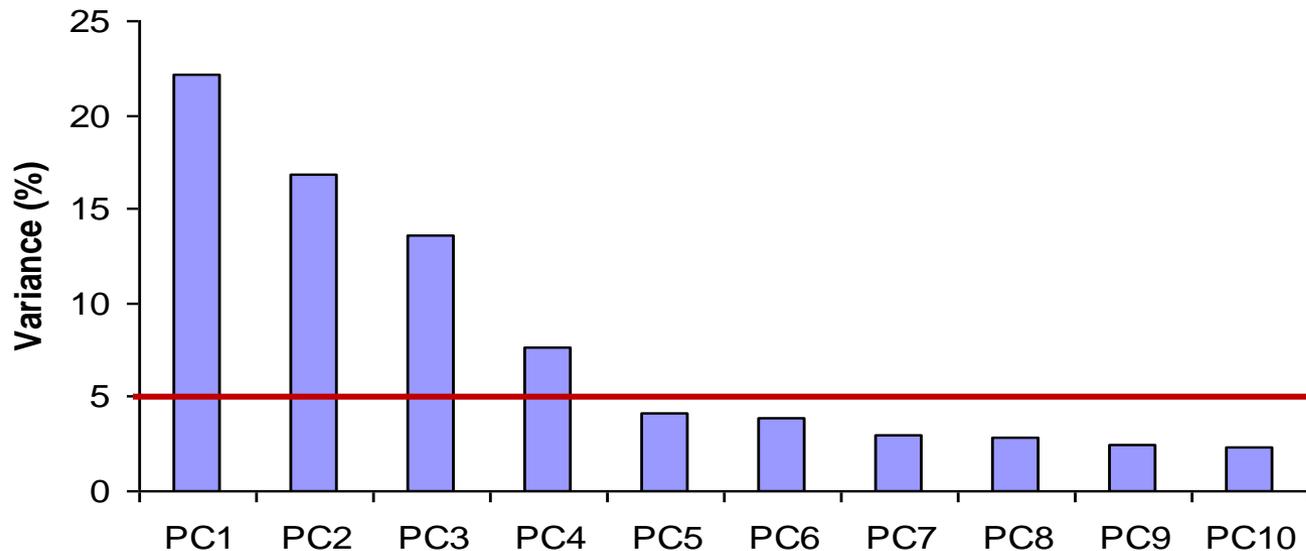


# Dimensionality Reduction using PCA

In high-dimensional problems, data sometimes lies near a linear subspace, as noise introduces small variability

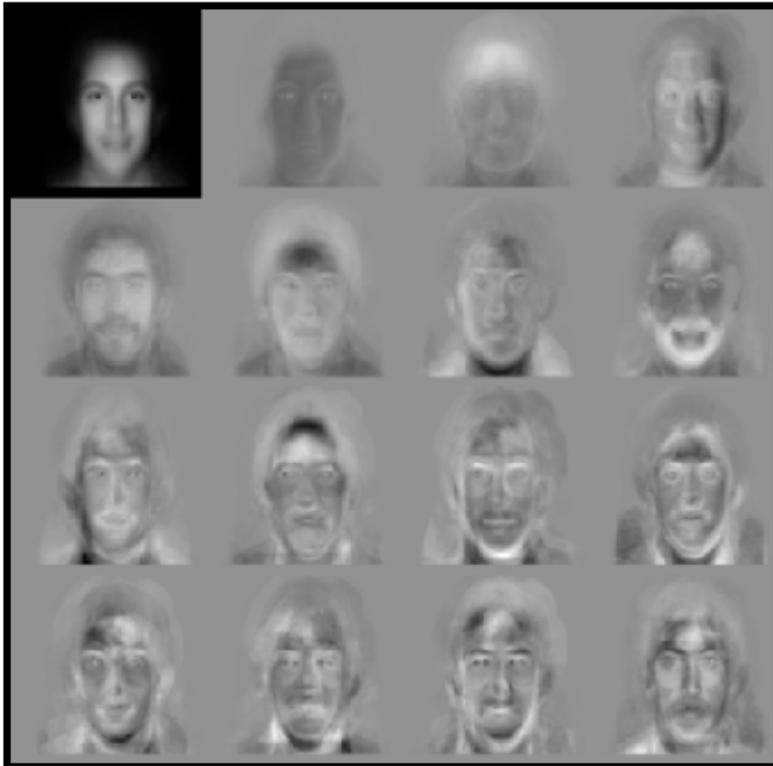
Only keep data projections onto principal components with **large** eigenvalues

Can **ignore** the components of smaller significance.



Might **lose some info**, but if eigenvalues are small, do not lose much

# Example: faces



**Eigenfaces**  
from 7562  
images:

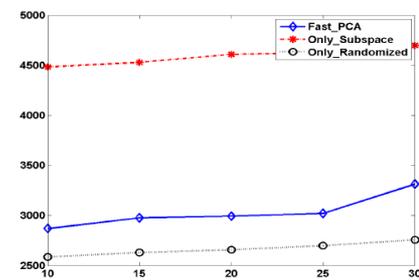
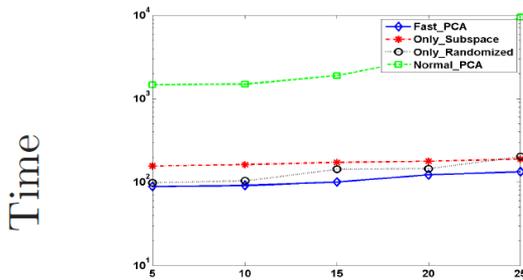
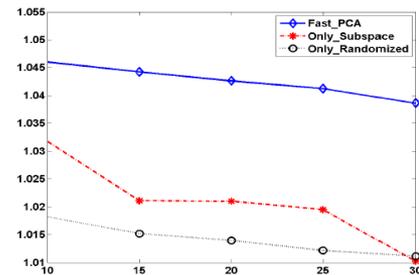
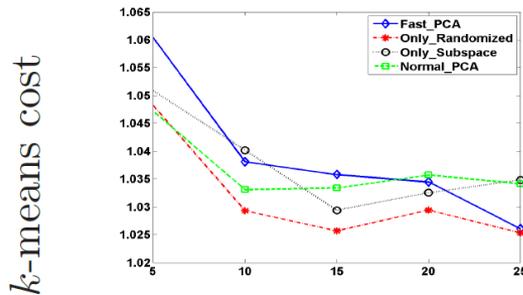
**top left image  
is linear  
combination  
of rest.**

Sirovich & Kirby (1987)  
Turk & Pentland (1991)

Can represent a face image using just 15 numbers!

- PCA provably useful before doing k-means clustering and also empirically useful. E.g.,

- ▷ **Performance:** cost increase < 5%;  $\times 10$  to  $\times 100$  speedup
- ▷ **k-Means Clustering:** *k*-means cost/time vs dimension



NewsGroups

BOWpubmed

# PCA Discussion

## Strengths

Eigenvector method

No tuning of the parameters

No local optima

## Weaknesses

Limited to second order statistics

Limited to linear projections

# Kernel PCA (Kernel Principal Component Analysis)

Useful when data lies on or near a low  $d$ -dimensional linear subspace of the  $\phi$ -space associated with a kernel

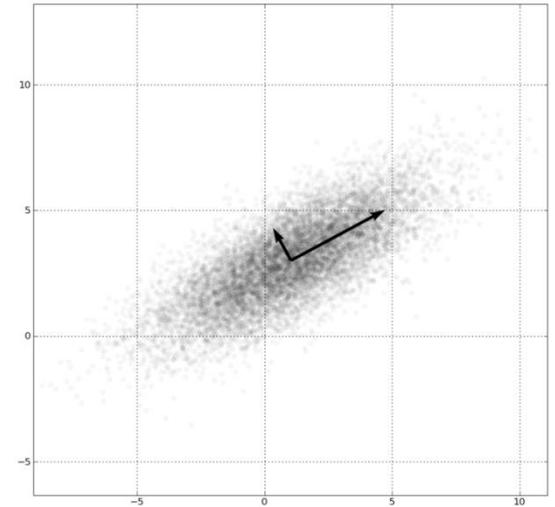
# Properties of PCA

- Given a set of  $n$  centered observations  $x_i \in R^D$ , 1<sup>st</sup> PC is the direction that maximizes the variance

- $X = (x_1, x_2, \dots, x_n)$

- $v_1 = \operatorname{argmax}_{\|v\|=1} \frac{1}{n} \sum_i (v^\top x_i)^2$   
 $= \operatorname{argmax}_{\|v\|=1} \frac{1}{n} v^\top X X^\top v$

- Covariance matrix  $C = \frac{1}{n} X X^\top$
- $v_1$  can be found by solving the eigenvalue problem:
  - $C v_1 = \lambda v_1$  (of maximum  $\lambda$ )

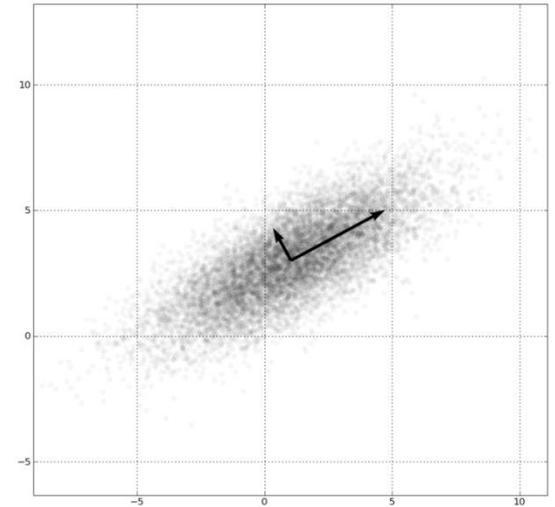


# Properties of PCA

- Given a set of  $n$  centered observations  $x_i \in R^D$ , 1<sup>st</sup> PC is the direction that maximizes the variance

- $X = (x_1, x_2, \dots, x_n)$

- $v_1 = \operatorname{argmax}_{\|v\|=1} \frac{1}{n} \sum_i (v^\top x_i)^2$   
 $= \operatorname{argmax}_{\|v\|=1} \frac{1}{n} v^\top X X^\top v$



- Covariance matrix  $C = \frac{1}{n} X X^\top$  is a  $D \times D$  matrix  
the  $(i, j)$  entry of  $X X^\top$  is the correlation of the  $i$ -th coordinate of examples with  $j$ th coordinate of examples
- To use kernels, need to use the inner-product matrix  $X^\top X$ .

# Alternative expression for PCA

- The principal component lies in the span of the data

$$v_1 = \sum_i \alpha_k x_i = X\alpha$$

**Why?** 1<sup>st</sup> PC is direction of largest variance, and for any direction outside of the span of the data, only get more variance if we project that direction into the span.

- Plug this in we have

$$Cv_1 = \frac{1}{n}XX^T X\alpha = \lambda X\alpha$$

- Now, left-multiply the LHS and RHS by  $X^T$ .

$$\frac{1}{n}X^TXX^T X\alpha = \lambda X^T X\alpha$$

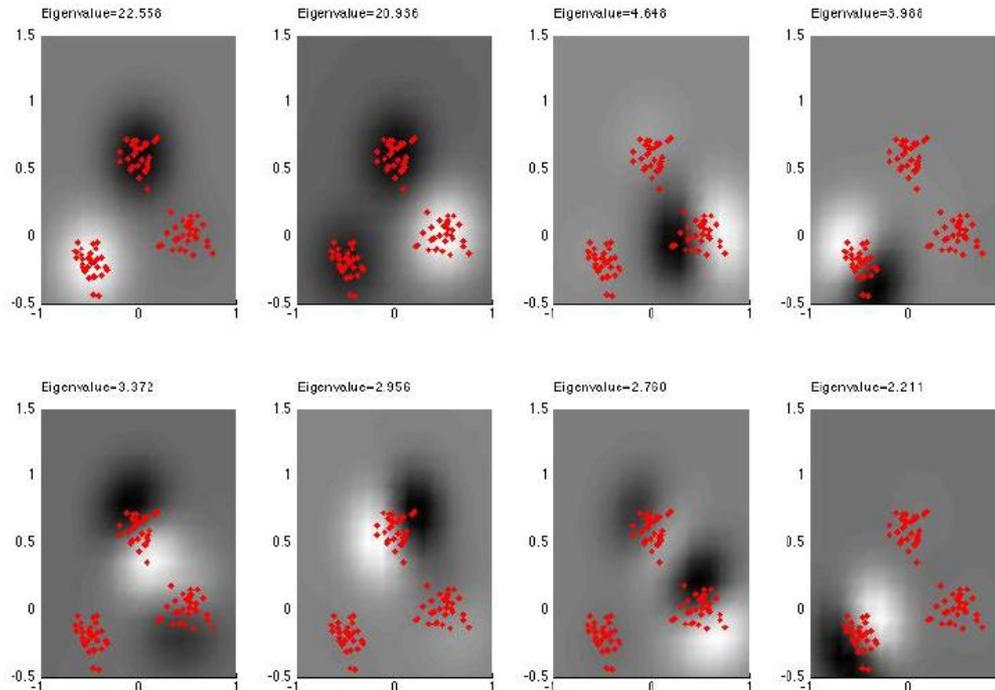
Only depends on  
the inner product  
matrix

# Kernel PCA

- **Key Idea:** Replace inner product matrix by kernel matrix
  - PCA:  $\frac{1}{n} X^T X X^T X \alpha = \lambda X^T X \alpha$
  - Let  $K = [K(x^i, x^j)]_{ij}$  be the matrix of all dot-products in the  $\phi$ -space.
  - Kernel PCA: replace " $X^T X$ " with  $K$ .  
 $\frac{1}{n} K K \alpha = \lambda K \alpha$ , or equivalently,  $\frac{1}{n} K \alpha = \lambda \alpha$
- **Key computation:** form an  $n$  by  $n$  kernel matrix  $K$ , and then perform eigen-decomposition on  $K$ .

# Kernel PCA Example

- Gaussian RBF kernel  $\exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$  over 2 dimensional space
- Eigenvector evaluated at a test point  $x$  is a function  
 $w^\top \phi(x) = \sum_i \alpha_i \langle \phi(x^i), \phi(x) \rangle = \sum_i \alpha_i k(x^i, x)$



# What You Should Know

- Principal Component Analysis (PCA)
  - What PCA is, what is useful for.
  - Both the maximum variance subspace and the minimum reconstruction error viewpoint.
- Kernel PCA

Additional material on computing the principal components and ICA

# Power method for computing PCs

Given matrix  $X \in R^{D \times n}$ , compute the top eigenvector of  $X X^T$

Initialize with random  $\hat{v} \in R^D$

**Repeat**

$$\hat{v} \leftarrow X X^T \hat{v}$$

$$\hat{v} \leftarrow \hat{v} / \|\hat{v}\|$$

**Claim**

For any  $\epsilon > 0$ , whp over choice of initial vector, after  $O\left(\frac{1}{\epsilon} \log \frac{d}{\epsilon}\right)$  iterations, we have  $\hat{v}^T X X^T \hat{v} \geq (1 - \epsilon)\lambda_1$ .

Then can subtract the  $\hat{v}$  component off of each example and repeat to get the next.

# Eigendecomposition

Any symmetric matrix  $A = XX^T$  is guaranteed to have an eigendecomposition with real eigenvalues:  $A = V \Lambda V^T$ .

$$\begin{array}{ccccccc} \boxed{\phantom{A}} & = & \boxed{\phantom{V}} & \boxed{\begin{array}{ccc} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_3 \\ & & \dots \end{array}} & \boxed{\phantom{V^T}} & = & \sum_i \lambda_i v_i v_i^T \\ \mathbf{A} & & \mathbf{V} & \mathbf{\Lambda} & \mathbf{V}^T & & \\ (D \times D) & & (D \times D) & (D \times D) & (D \times D) & & \end{array}$$

Matrix  $\Lambda$  is diagonal with eigenvalues  $\lambda_1 \geq \lambda_2 \geq \dots$  on the diagonal. Matrix  $V$  has the eigenvectors as the columns.

# Singular Value Decomposition (SVD)

Eigendecomposition of  $XX^T$  is closely related to SVD of  $X$ .

Given a matrix  $X \in \mathbb{R}^{D \times n}$ , the SVD is a decomposition:  $X^T = USV^T$

$$\begin{array}{c} \boxed{\phantom{X^T}} \\ X^T \\ (n \times D) \end{array} = \begin{array}{c} \boxed{\phantom{U}} \\ U \\ (n \times d) \end{array} \begin{array}{c} \boxed{\begin{array}{cc} \sigma_1 & 0 \\ 0 & \sigma_2 \dots \end{array}} \\ S \\ (d \times d) \end{array} \begin{array}{c} \boxed{\phantom{V^T}} \\ V^T \\ (d \times D) \end{array} = \sum_i \sigma_i u_i v_i^T$$

- $S$  is a diagonal matrix with the singular values  $\sigma_1, \dots, \sigma_d$  of  $X$ .
- Columns of  $U, V$  are orthogonal, unit length.
- So,  $XX^T = VSU^TUSV^T = VS^2V^T =$  eigendecomposition of  $XX^T$ .

So,  $\lambda_i = \sigma_i^2$  and can read off the solution from the SVD.

# Singular Value Decomposition (SVD)

Eigendecomposition of  $XX^T$  is closely related to SVD of  $X$ .

Given a matrix  $X \in \mathbb{R}^{D \times n}$ , the SVD is a decomposition:  $X^T = USV^T$

$$\begin{array}{c} \boxed{\phantom{X^T}} \\ X^T \\ (n \times D) \end{array} = \begin{array}{c} \boxed{\phantom{U}} \\ U \\ (n \times d) \end{array} \begin{array}{c} \boxed{\begin{array}{cc} \sigma_1 & 0 \\ 0 & \sigma_2 \\ & \dots \end{array}} \\ S \\ (d \times d) \end{array} \begin{array}{c} \boxed{\phantom{V^T}} \\ V^T \\ (d \times D) \end{array} = \sum_i \sigma_i u_i v_i^T$$

- In fact, can view the rows of  $US$  as the coordinates of each example along the axes given by the  $d$  eigenvectors.

So,  $\lambda_i = \sigma_i^2$  and can read off the solution from the SVD.

# Independent Component Analysis (ICA)

Find a linear transformation

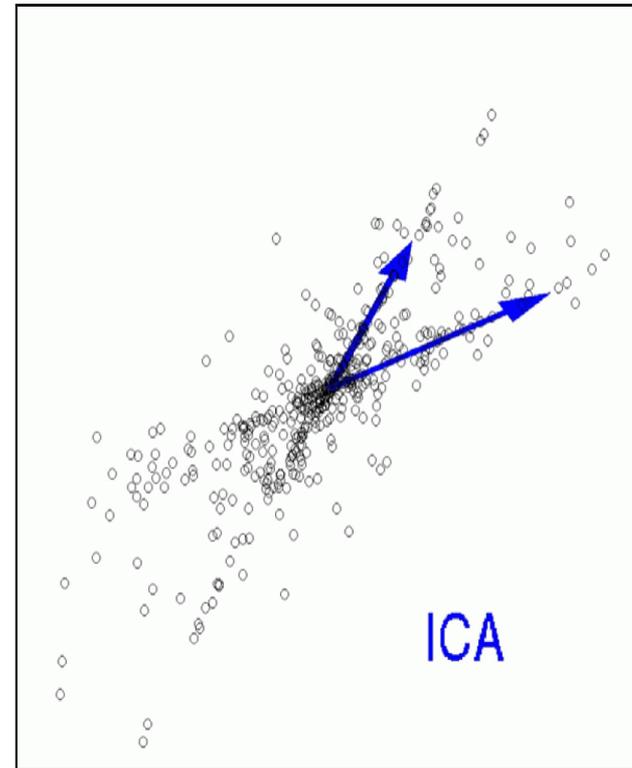
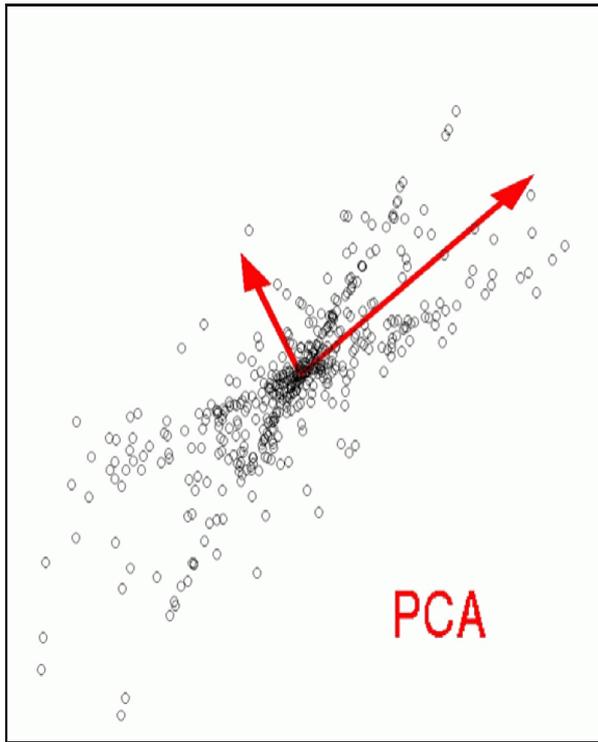
$$\mathbf{x} = \mathbf{V} \cdot \mathbf{s}$$

for which coefficients  $\mathbf{s} = (s_1, s_2, \dots, s_D)^T$  are **statistically independent**

$$p(s_1, s_2, \dots, s_D) = p_1(s_1)p_2(s_2) \dots p_n(s_D)$$

Algorithmically, we need to identify matrix  $\mathbf{V}$  and coefficients  $\mathbf{s}$ , s.t. under the condition  $\mathbf{x} = \mathbf{V}^T \cdot \mathbf{s}$  the **mutual information** between  $s_1, s_2, \dots, s_D$  is minimized:

$$I(s_1, s_2, \dots, s_D) = \sum_{i=1}^D H(s_i) - H(s_1, s_2, \dots, s_D)$$



PCA finds directions of maximum variation,  
ICA would find directions most "aligned" with data.