Feature Selection: Linear Transformations

$$\underline{y}_{new} = M\, \underline{x}_{old}$$

Constraint Optimization (insertion)

Problem: Given an objective function $f(x)$ to be optimized and let constraints be given by $h_k(x)=c_k$,
moving constants to the left, ==> $h_k(x) - c_k = g_k(x)$.
$f(x)$ and $g_k(x)$ must have continuous first partial derivatives

A Solution:

Lagrangian Multipliers $0 = \nabla_x f(x) + \Sigma \nabla_x \lambda_k\, g_k(x)$

or starting with the Lagrangian : $L(x,\lambda) = f(x) + \Sigma\, \lambda_k\, g_k(x)$.

with $\nabla_x L(x,\lambda) = 0$.

The Covariance Matrix (insertion)

Definition

Let $\underline{x} = \{x_1, \ldots, x_N\} \in \mathbb{R}^N$ be a real valued random variable (data vectors), with the expectation value of the mean $E[\underline{x}] = \mu$.

We define the **covariance matrix** Σ_x of a random variable $\underline{x}$

as $\Sigma_x := E[(\underline{x} - \mu)(\underline{x} - \mu)^T]$

with matrix elements $\Sigma_{ij} = E[(\underline{x}_i - \mu_i)(\underline{x}_j - \mu_j)^T]$.

Application: Estimating $E[\underline{x}]$ and $E[(\underline{x} - E[\underline{x}])(\underline{x} - E[\underline{x}])^T]$ from data.

We assume m samples of the random variable $\underline{x} = \{x_1, \ldots, x_N\} \in \mathbb{R}^N$
that is we have a set of m vectors $\{\underline{x}_1, \ldots, \underline{x}_m\} \in \mathbb{R}^N$
or when put into a data matrix $X \in \mathbb{R}^{N \times m}$

Maximum Likelihood estimators
for μ and Σ_x are:

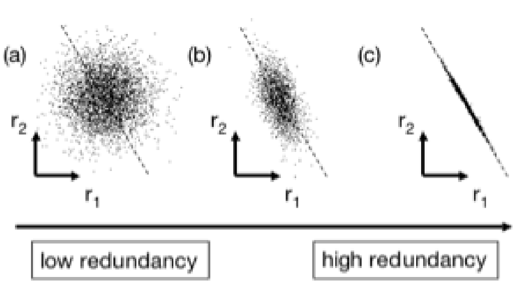
$$\Rightarrow \quad \underline{\mu}_{ML} = \frac{1}{m} \sum_{k=1}^{m} \underline{x}_k$$

$$\Rightarrow \quad \Sigma_{ML} = \frac{1}{m} \sum_{k=1}^{m} (\underline{x}_k - \underline{\mu}_{ML})(\underline{x}_k - \underline{\mu}_{ML})^T \quad = \frac{1}{m} X X^T$$

KLT/PCA Motivation

5

- Find meaningful "directions" in correlated data
- Linear dimensionality reduction
- Visualization of higher dimensional data
- Compression / Noise reduction
- PDF-Estimate



Karhunen-Loève Transform: 1st Derivation

Problem

Let $\underline{x} = \{x_1, ..., x_N\} \in \mathbb{R}^N$ be a feature vector of zero mean, real valued random variables.

We seek the direction $\underline{a}_1$ of maximum variance:

$==>$ $y_1 = \underline{a}_1^T \underline{x}$ for which $\underline{a}_1$ is such as $E[y_1^2]$ is maximum with the constraint that $\underline{a}_1^T \underline{a}_1 = 1$

This is a constrained optimization → use of the Lagrangian:

$$L(\underline{a}_1, \lambda_1) = E[\underline{a}_1^T \underline{x}\, \underline{x}^T \underline{a}_1] - \lambda_1 (\underline{a}_1^T \underline{a}_1 - 1)$$

$$= \underline{a}_1^T \Sigma_x \underline{a}_1 - \lambda_1 (\underline{a}_1^T \underline{a}_1 - 1)$$

Lagrange multiplier

Karhunen-Loève Transform

$$L(\underline{a}_1, \lambda_1) = \underline{a}_1^T \Sigma_x \underline{a}_1 - \lambda_1 (\underline{a}_1^T \underline{a}_1 - 1)$$

for $E[y_1^2]$ to be maximum : $\dfrac{\partial L(\underline{a}_1, \lambda_1)}{\partial \underline{a}_1} = 0$

$=>$ $\Sigma_x \underline{a}_1 - \lambda_1 \underline{a}_1 = 0$

$=>$ $\underline{a}_1$ must be *eigenvector* of Σ_x with *eigenvalue* λ_1.

$E[y_1^2] = \underline{a}_1^T \Sigma_x \underline{a}_1 = \lambda_1$

$=>$ for $E[y_1^2]$ to be maximum, λ_1 must be the *largest* eigenvalue.

Karhunen-Loève Transform

Now let's search for a second direction, $\underline{\boldsymbol{a}}_2$, such that:

$y_2 = \underline{\boldsymbol{a}}_2^{\mathrm{T}} \underline{\mathbf{x}}$ such as $\mathrm{E}[y_2^{\,2}]$ is maximum, and

$\underline{\boldsymbol{a}}_2^{\mathrm{T}} \underline{\boldsymbol{a}}_1 = 0$ and $\underline{\boldsymbol{a}}_2^{\mathrm{T}} \underline{\boldsymbol{a}}_2 = 1$

Similar derivation: $\mathrm{L}(\underline{\boldsymbol{a}}_2, \lambda_2) = \underline{\boldsymbol{a}}_2^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{x}} \underline{\boldsymbol{a}}_2 - \lambda_2 \, (\underline{\boldsymbol{a}}_2^{\mathrm{T}} \underline{\boldsymbol{a}}_2 - 1)$ with $\underline{a}_2^{\mathrm{T}} \underline{a}_1 = 0$

=> $\underline{\boldsymbol{a}}_2$ must be the *eigenvector* of $\boldsymbol{\Sigma}_{\mathbf{x}}$ associated with the *second largest eigenvalue* λ_2.

We can derive $\boldsymbol{N}$ orthonormal directions that maximize the variance: $\boldsymbol{A} = [\underline{\boldsymbol{a}}_1, \underline{\boldsymbol{a}}_2, \ldots, \underline{\boldsymbol{a}}_N]$ and $\underline{\mathbf{y}} = \boldsymbol{A}^{\mathrm{T}} \underline{\mathbf{x}}$

The resulting matrix $\boldsymbol{A}$ is known as Principal Component Analysis (PCA)

or Kharunen-Loève transform (KLT) $\underline{\mathbf{y}} = \boldsymbol{A}^{\mathrm{T}} \underline{\mathbf{x}}$ $\underline{\mathbf{x}} = \sum_{i=1}^{N} y_i \underline{\boldsymbol{a}}_i$

Karhunen-Loève Transform: 2nd Derivation

Problem

Let $\underline{\mathbf{x}} = \{x_1, \ldots, x_N\} \in \mathbb{R}^{\mathrm{N}}$ be a feature vector of ~~zero mean~~, real valued random variables.

We seek a transformation $\boldsymbol{A}$ of $\underline{\mathbf{x}}$ that results in a new set of variables $\underline{y} = \boldsymbol{A}^{\mathrm{T}} \underline{x}$ (feature vectors) which are uncorrelated (i.e. $\mathrm{E}[y_i\, y_j] = 0$ for $i \neq j$) .

- Let $\underline{\mathbf{y}} = A^T \underline{\mathbf{x}}$, then by definition of the correlation matrix:

$$\boxed{R_y \equiv E[\boldsymbol{y}\boldsymbol{y}^T] = E[A^T \boldsymbol{x}\boldsymbol{x}^T A] = A^T R_x A}$$

- R_x is symmetric $\Rightarrow$ its eigenvectors are mutually orthogonal

Karhunen-Loève Transform

- i.e. if we choose A such that its columns a_i are orthonormal eigenvectors of R_x , we get:

$$R_y = A^T R_x A = \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_N \end{bmatrix}$$

- If we further assume R_x to be positive definite,

  ---- > the eigenvalues λ_i will be positive.

The resulting matrix $\boldsymbol{A}$ is known as

Karhunen-Loève transform (KLT) $\underline{\boldsymbol{y}} = A^T \underline{\boldsymbol{x}}$ $\underline{\boldsymbol{x}} = \sum_{i=1}^{N} y_i \underline{\boldsymbol{a}}_i$

Karhunen-Loève Transform

The Karhunen-Loève transform (KLT)

$$\underline{y} = \boldsymbol{A}^T \underline{x} \qquad \underline{\boldsymbol{x}} = \sum_{i=1}^{N} y_i \underline{\boldsymbol{a}}_i$$

For mean-free vectors (e.g. replace $\underline{\boldsymbol{x}}$ *by* $\underline{\boldsymbol{x}} - E[\,\underline{\boldsymbol{x}}\,]$)

this process diagonalizes the covariance matrix $\boldsymbol{\Sigma}_y$

KLT Properties: MSE-Approximation

We define a new vector $\hat{x}$ in m-dimensional subspace ($m < N$),

using only m basis vectors: $\boxed{\hat{\mathbf{x}} = \sum_{i=1}^{m} y_i \mathbf{a}_i}$

- Projection of $\underline{x}$ into the subspace spanned by the m used (orthonormal) eigenvectors.

Now, what is the expected mean square error between $\mathbf{x}$ and its projection $\hat{x}$:

$$E\left[\left\|\mathbf{x} - \hat{\mathbf{x}}\right\|^2\right] = E\left[\left\|\sum_{i=m+1}^{N} y_i \mathbf{a}_i\right\|^2\right] = E\left[\sum_i \sum_j (y_i \mathbf{a}_i^T)(y_j \mathbf{a}_j)\right] = \cdots$$

KLT Properties: MSE-Approximation

$$E\left[\left\|\mathbf{x} - \hat{\mathbf{x}}\right\|^2\right] = ... = E\left[\sum_i \sum_j (y_i \mathbf{a}_i^T)(y_j \mathbf{a}_j)\right] = \sum_{i=m+1}^{N} E\left[y_i^2\right] = \sum_{i=m+1}^{N} \lambda_i$$

The error is minimized if we choose as basis those eigenvectors corresponding to the $\boldsymbol{m}$ largest eigenvalues of the correlation matrix.

- Amongst all other possible orthogonal transforms KLT is the one leading to minimum MSE

This form of KLT (applied to mean free data) is also referred to as **Principal Component Analysis** (PCA).
The principal components are the eigenvectors ordered (desc.) by their respective eigenvalue magnitudes λ_i

KLT Properties

15

Total variance

- Let w.l.o.g. $\mathrm{E}[\boldsymbol{x}]=0$ and $\boldsymbol{y} = A^T\boldsymbol{x}$ the KLT (PCA) of $\boldsymbol{x}$. From the previous definitions we get:

$$\boxed{\sigma_{y_i}^2 \equiv E\left[y_i^2\right] = \lambda_i}$$

- i.e. the eigenvalues of the input covariance matrix are equal to the variances of the transformed coordinates.

> ➢ Selecting those features corresponding to m largest eigenvalues retains the maximal possible total variance (sum of component variances) associated with the original random variables x_i.

KLT Properties: Entropy

16

For a random vector $\boldsymbol{y}$ the entropy $H_y = -E[\ln p_y(\boldsymbol{y})]$ is a measure for the randomness of the underlying process.

Example: for a zero-mean (μ=0) m-dim. Gaussian

$$H_y = -E\left[\ \ln(\ (2\pi)^{-\frac{m}{2}} \left|\Sigma_y\right|^{-\frac{1}{2}} \exp(-\frac{1}{2}\underline{y}^{\mathrm{T}}\Sigma_y^{-1}\underline{y})\ )\ \right]$$

$$H_y = \tfrac{m}{2}\ln(2\pi) + \tfrac{1}{2}\ln\left|\Sigma_y\right| + \tfrac{1}{2}E[\boldsymbol{y}^T\Sigma_y^{-1}\boldsymbol{y}]$$

$$= \tfrac{m}{2}\ln(2\pi) + \tfrac{1}{2}\ln\prod_{i=1}^{m}\lambda_i + \frac{m}{2}$$

$$E[\boldsymbol{y}^T\Sigma_y^{-1}\boldsymbol{y}] = E[trace\left\{\boldsymbol{y}^T\Sigma_y^{-1}\boldsymbol{y}\right\}]$$
$$= E[trace\left\{\Sigma_y^{-1}\boldsymbol{y}\boldsymbol{y}^T\right\}]$$
$$= E[trace\left\{I\right\}] = m$$

- ➢ Selecting those features corresponding to m largest eigenvalues maximizes the entropy in the remaining features.
- ➢ No wonder: variance and randomness are directly related !

Computing a PCA:

<u>Problem:</u> Given mean free data X , a set on n feature vectors $x_i \in R^m$. Compute the orthonormal eigenvectors a_i of the correlation matrix R_x .

- There are many algorithms that can compute very efficiently eigenvectors of a matrix.
  However, most of these methods can be very unstable in certain special cases.

- Here we present SVD, a method that is in general not the most efficient one.
  However, the method can be made numerically stable very easily!

Computing a PCA:

Singular Value Decomposition:

an Excursus to Linear Algebra
(without Proofs)

19

Singular Value Decomposition :

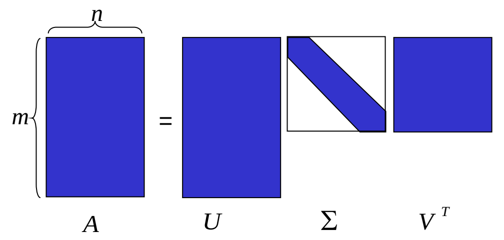
<u>SVD (reduced Version)</u>: For matrices $A \in R^{m\times n}$ with $m \geq n$, there exist matrices

$U \in R^{m\times n}$ with orthonormal columns ($U^{\mathrm{T}}U = I$) ,

$V \in R^{n\times n}$ orthogonal ($V^{\mathrm{T}}V = I$),

$\Sigma \in R^{n\times n}$ diagonal,

with $A = U\,\Sigma\,V^{\mathrm{T}}$



- The diagonal values of Σ (σ_1, σ_2, …., σ_n) are called the singular values.
- It is accustomed to sort them: $\sigma_1 \geq \sigma_2 \geq \ldots. \geq \sigma_n$

20

SVD Applications:

SVD is an all-rounder !

Once you have U, Σ, V, you can use it to:

- Solve Linear Systems: $\boldsymbol{A}\,\underline{x} = \underline{b}$
  - a) If A^{-1} exists → Compute matrix inverse
  - b) for fewer equations than unknowns
  - c) for more equations than unknowns
  - d) if there is no solution: compute

    x that $|\boldsymbol{A}\,\underline{x} - \underline{b}| = min$
  - e) compute rank (numerical rank) of a matrix
- …….
- Compute PCA / KLT

SVD : Matrix inverse A^{-1}

$\boldsymbol{A}\,\underline{x} = \underline{b}$:

$A = U\,\Sigma\,V^{T}$ $\qquad$ $U, \Sigma, V,$ exist for all A

If A is square nxn and not singular, then A^{-1} exists.

$$A^{-1} = \left(U\,\Sigma\,V^{T}\right)^{-1}$$

$$= \left(V^{T}\right)^{-1}\Sigma^{-1}U^{-1}$$

$$= V \begin{bmatrix} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_n} \end{bmatrix} U^{T}$$

Computing A^{-1} for a singular A !?

Since U, Σ, V all exist, the only problem can originate if one $\sigma_i = 0$ or numerically close to zero.

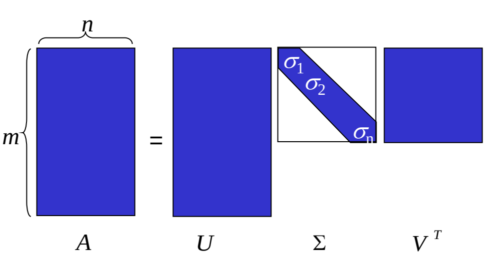
--> singular values indicate if A is singular or not!!

SVD : Rank of a Matrix

- The *rank* of A is the number of non-zero singular values.
- If there are very small singular values σ_i, then A is close of being singular.

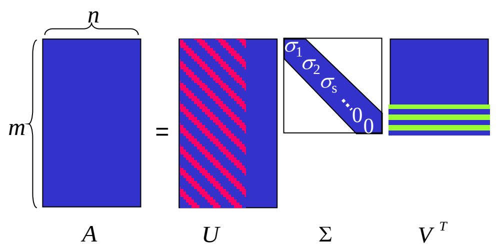
We can set a threshold t, and set $\sigma_i = 0 \;\; if \;\; \sigma_i \leq t$

then the $\quad numeric\_rank\,(A) = \#\{\,\sigma_i \mid \sigma_i > t\,\}$



SVD : Rank of a Matrix (2)

- numeric_rank(A) = # { σ_i | $\sigma_i > t$ } ,
  the rank of A is equal the $dim(Img(A))$

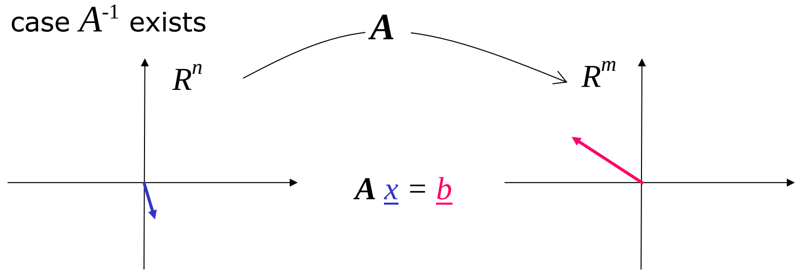


$n = dim(Img(A)) + dim(Ker(A))$

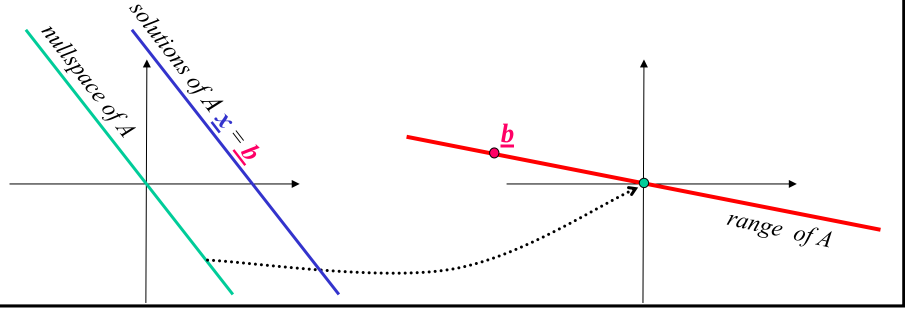
- the columns of U corresponding to the $\sigma_i \neq 0$, span the range of A
- the columns of V corresponding to the $\sigma_i = 0$, span the nullspace of A

remember linear mappings $A\, \underline{x} = \underline{b}$

1) case A^{-1} exists

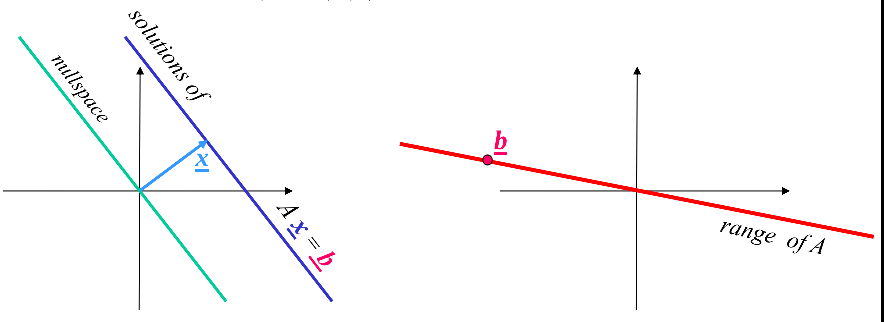


2) A is singular: $dim(Ker(A)) \neq 0$



SVD : solving $\boldsymbol{A}\,\underline{x} = \underline{b}$

2) A is singular: $dim(\, Ker(A)\,) \neq 0$



There are an infinite number of different $\underline{x}$ that solve $A\underline{x}=\underline{b}$!!??

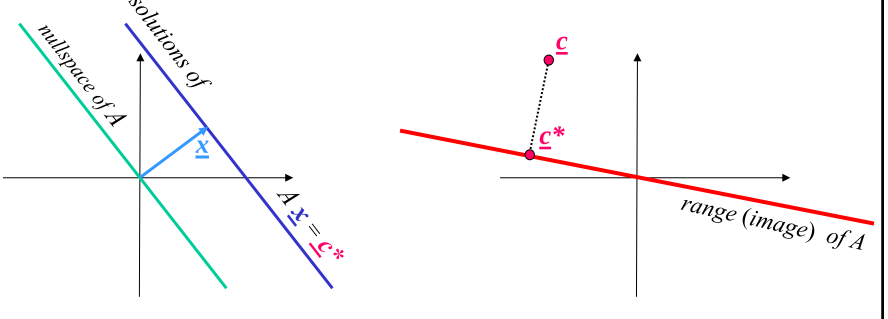
Which one should we choose??

e.g. we can choose the $\underline{x}$ with $\|\,\underline{x}\,\| = min$

$\rightarrow$ then we have to search in the space *orthogonal* to the *nullspace*

SVD : Solving $\|\,\boldsymbol{A}\,\underline{x} - \underline{c}\,\| = min$

3) $\underline{c}$ is not in the range of A



1) Projecting $\underline{c}$ into the *range* of A results in $\underline{c}^*$

2) From all the solutions of $A\,\underline{x} = \underline{c}^*$ we choose the $\underline{x}$ with $\|\,\underline{x}\,\| = min$

SVD : Solving $\| \boldsymbol{A} \, \underline{x} - \underline{c} \| = min$

$\boldsymbol{A} \, \underline{x} = \underline{c}$ for any A exist U, Σ, V, with $A = U \Sigma V^T$
with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$

$$U \Sigma V^T \underline{x} = \underline{c}$$

$$\underline{x} = \left(U \Sigma V^T\right)^{-1} \underline{c}$$

$$= \left(V^T\right)^{-1} \Sigma^{-1} U^{-1} \underline{c}$$

$$= V \begin{bmatrix} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_n} \end{bmatrix} U^T \underline{c}$$

Computing A^{-1} for a singular A !?

--> What to do in Σ^{-1} with $1/0 = \infty$????

Some $\sigma_i = 0$ *if* $\sigma_i \leq t$

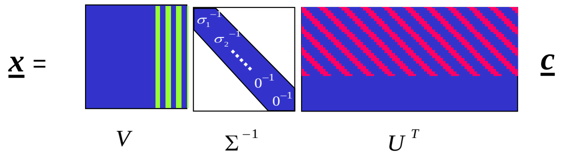
Remember what we need ---- >

SVD : Solving: $\| \boldsymbol{A} \, \underline{x} - \underline{c} \| = min$

We need to:

1) Project $\underline{c}$ into the *range* of A to obtain a $\underline{c}^*$

2) From all the solutions of $A \, \underline{x} = \underline{c}^*$ we choose the $\| \underline{x} \| = min$
that is the $\underline{x}$ in the space *orthogonal* to the *nullspace*



- the columns of U corresponding to the $\sigma_i \neq 0$, span the range of A
- the columns of V corresponding to the $\sigma_i = 0$, span the nullspace of A

Basically all rows or columns multiplied by 1/0 are irrelevant!!

--> so even setting 1/0 = 0 , will lead to the correct result.

SVD at Work:

For Linear Systems $\mathbf{A}\,\underline{x} = \underline{b}$:

Case fewer equations than unknowns:

→ fill rows of A with zeros so that $n = m$

Perform SVD on $\mathbf{A}$ with $(n \leq m)$:

→ Compute U, Σ, V, with $A = U\,\Sigma\,V^{T}$

→ Compute threshold t and

→ in Σ set $\sigma_i = 0$ *for all* $\sigma_i \leq t$

→ in Σ^{-1} set $1/\sigma_i = 0$ *for all* $\sigma_i \leq t$

For Linear Systems: compute Pseudoinverse $A^{+} = V\,\Sigma^{-1}\,U^{T}$
and compute $\underline{x} = A^{+}\,\underline{b}$

Application: Compute PCA via SVD

Problem: Given mean free data X , a set on n feature vectors $\underline{x}_i \in R^m$ compute the orthonormal eigenvectors a_i of the correlation matrix R_x .

Now we use SVD

1. Move center of mass to origin: $x_i' = x_i - \mu$
2. Build data matrix, from mean free data $X = U\,\Sigma\,V^{T}$
3. The principal axes are eigenvector of the covariance matrix $C = 1/n\ XX^{T}$

$$XX^{T} = U \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{bmatrix} U^{T}$$

Application: Compute PCA via SVD (2)

31

with SVD

$$XX^T = U \Sigma V^T (U \Sigma V^T)^T$$
$$= U \Sigma V^T (V \Sigma^T U^T)$$
$$= U \Sigma \Sigma^T U^T$$
$$= U \Sigma^2 U^T$$

Since $C = 1/n \; XX^T$

the eigenvalues compute to $\lambda_i = 1/n \; \sigma_i^2$

σ from SVD

with $\lambda_i = \sigma_i^2$

σ^2 variance of $E[y_i^2]$

Example: PCA on Images

32

- Assume we have a set of k images (of size $N \times N$)
- Each image can be seen as N^2-dimensional point p_i (lexicographically ordered); the whole set can be stored as matrix:

$$X = \begin{bmatrix} | & | & & | \\ \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_k \\ | & | & & | \end{bmatrix}$$

- Computing PCA the "naïve" way
  - Build correlation matrix XX^T (N^4 elements)
  - Compute eigenvectors from this matrix: $O((N^2)^3)$

➢ Already for small images (e.g. $N=100$) this is far too expensive

PCA on Images

33

Now we use SVD

1. Move center of mass to origin: $p_i' = p_i - \mu$
2. Build data matrix, from mean free data $$X = \begin{bmatrix} | & | & & | \\ \mathbf{p}_1' & \mathbf{p}_2' & \cdots & \mathbf{p}_n' \\ | & | & & | \end{bmatrix}$$
3. The principal axes are eigenvector of

$$XX^T = U \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{bmatrix} U^T$$

PCA on Images

34

Faces

mean face

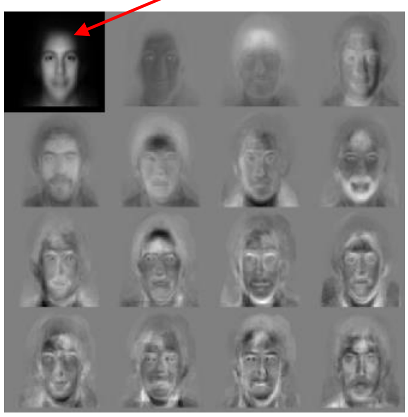
Eigenfaces

Principal Components can be visualized by adding to the mean vector an eigenvector multiplied by a factor (e.g. λ)

PCA applied to face images

Here the faces where normalized in eye distance and eye position.

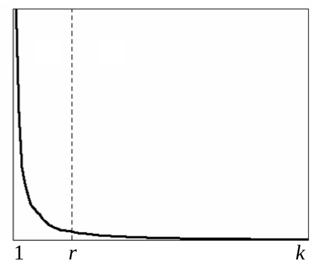
mean
face



Eigenfaces

Choosing subspace dimension *r*:

- Look at decay of the eigenvalues as a function of *r*
- Larger *r* means lower expected error in the subspace data approximation

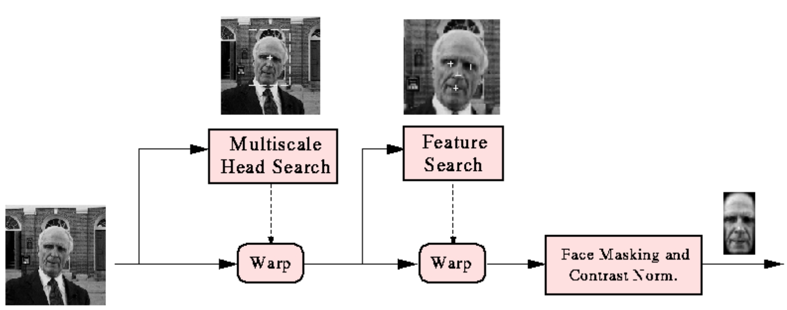


Eigenvalue spectrum

Eigenfaces for Face Recognition

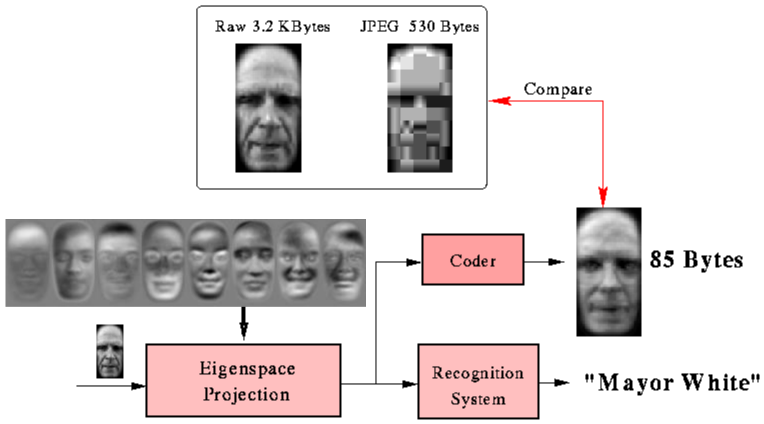
In the 90's the best performing Face Recognition System!

Turk, M. and Pentland, A. (1991).
Face recognition using eigenfaces.
In Proceedings of Computer Vision and Pattern Recognition, pages 586--591. IEEE.



37

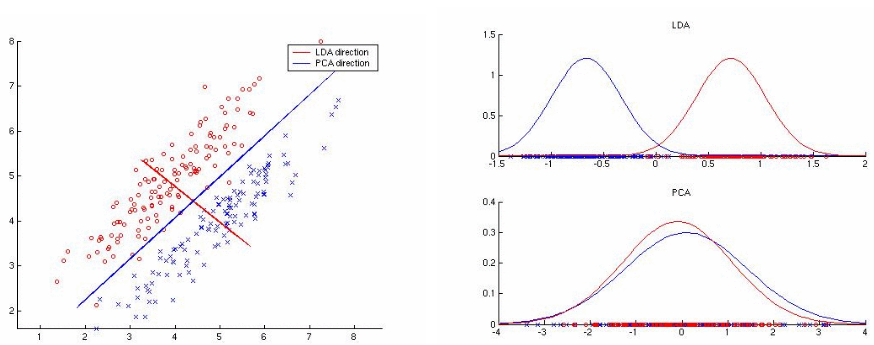
PCA for Face Recognition



38

PCA & Discrimination

- PCA/KLT do not use any class labels in the construction of the transform.
  - ➢ The resulting features may obscure the existence of separate groups.



PCA Summary

- Unsupervised: no assumption about the existence or nature of groupings within the data.
- PCA is similar to learning a Gaussian distribution for the data.
- Optimal basis for compression (if measured via MSE).
- As far as dimensionality reduction is concerned this process is distribution-free, i.e. it's a mathematical method without underlying statistical model.
- Extracted features (PCs) often lack 'intuition'.

PCA an Neural Networks

A three-layer NN with linear hidden units, trained as auto-encoder, develops an internal representation that corresponds to the principal components of the full data set. The transformation F1 is a linear projection onto a k-dimensional (*Duda, Hart and Stork:* chapter 10.13.1).