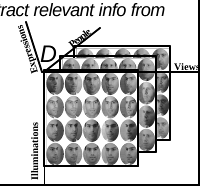
Lecture 2:
Feature and Model Selection

PCA and ICA

Statistical Learning

- Statistics: the science of collecting, organizing, and interpreting *data.* Machine learning using statistics
  - *Data collection.*
  - *Data analysis - organize & summarize data to bring out main features and clarify their underlying structure.*
  - *Inference and decision theory – extract relevant info from collected data and use it as a guide for further action.*



Designing a Machine Learning System

Data
↓
Feature and Model Selection
↓
Learning
↓
Evaluation

Data

- Data Collection:
  - Causation, Common Response, Confounding
  - Designing a Randomized Comparative Experiment
- Data may need a lot of:
  - Cleaning
  - Preprocessing (conversions)
- Cleaning:
  - Get rid of errors, noise,
  - Removal of redundancies
- Pre-processing:
  - Mean
  - Rescaling - continuous values transformed to some range, typically [-1, 1] or [0,1]

Feature Selection

- The size (dimensionality) of a sample can be enormous
- Example: document classification
  - 10,000 different words
  - Inputs: counts of occurrences of different words
  - Too many parameters to learn (not enough samples to justify the estimates the parameters of the model)
- Dimensionality reduction: replace inputs with features
  - Extract relevant inputs (e.g. mutual information measure)
  - **PCA – principal component analysis**
  - Group (cluster) similar words (uses a similarity measure)
- Replace with the group label

Model Selection

What is the right model to learn?

- A prior knowledge helps a lot, but still a lot of guessing
- Initial data analysis and visualization
  - We can make a good guess about the **form of the distribution**, **shape of the function**
- Independences and correlations
- Overfitting problem
  - Take into account the bias and variance of error estimates

Learning

- Learning - Optimization problem
  - Optimization problems can be hard to solve.
    Model and error function choice make a difference.
  - Parameter optimizations
    - Gradient descent, Conjugate gradient
    - Newton-Rhapson
    - Levenberg - Marquard
    - Some can be carried on-line on a sample by sample basis
    - Combinatorial optimizations (over discrete spaces):
      - Hill-climbing
      - Simulated-annealing
      - Genetic algorithms

Evaluation

- Problem: we cannot be 100 % sure about generalization
- Solution: test the statistical significance of the result

PCA

The Principle Behind Principal Component Analysis[1]

- Also called: - Hotteling Transform[2] or the
  - Karhunen-Loeve Method [3].
- *Find an orthogonal coordinate system such that data is approximated best and the correlation between different axis is minimized.*

1 I.T.Jolliffe; Principle Component Analysis; 1986
2 R.C.Gonzalas, P.A.Wintz; Digital Image Processing; 1987
3 K.Karhunen; Uber Lineare Methoden in der Wahrscheinlichkeits Rechnug; 1946
M.M.Loeve; Probability Theory; 1955

Assumptions

- The relationship between explanatory and response variable is linear
- Data has a gaussian distribution

PCA Goal

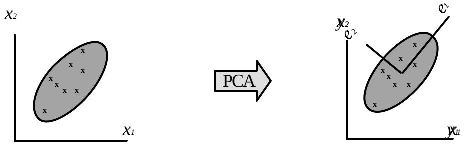
Problem Statement:

- Input: $\mathbf{X}=[\mathbf{x}_1|\ldots|\mathbf{x}_N]_{dxN}$
  N points in d-dimensional space
- Look for: $\mathbf{U}$, a $d \times m$ transformation matrix that maps $\mathbf{X}$ from d-dimensional space to m-dimensional space where ($m \leq d$).

st. $[\mathbf{y}_1|\ldots|\mathbf{y}_N]_{mxN} = \mathbf{U}^T\,[\mathbf{x}_1|\ldots|\mathbf{x}_N]$

& the covariance is minimized

PCA: Theory



- Define a new origin as the mean of the data set
- Find the direction of maximum variance in the samples (e_1) and align it with the first axis (y_1),
- Continue this process with orthogonal directions of decreasing variance, aligning each with the next axis
- Thus, we have a rotation which minimizes the covariance.

PCA: The Covariance Matrix

- Define the covariance (scatter) matrix of the input samples as :

$$\mathrm{Cov}(\mathbf{x}) = \mathbf{S}_T = \sum_{k=1}^{N} (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T$$

(where m is the sample mean)

- Let $\mathbf{D} = [\mathbf{x}_1\text{-m}, \ldots, \mathbf{x}_N\text{-m}]$ then the above expression can be rewritten simply as :

$$\mathrm{Cov}(\mathbf{x}) = \mathbf{S}_T = \mathbf{D}\mathbf{D}^T$$

Covariance Matrix Properties

- The matrix Cov is symmetric and of dimension $d \times d$.
- The diagonal contains the variance of each parameter (i.e. element Cov_{ii} is the variance in the i'th direction).
- Each element Cov_{ij} is the co-variance between the two directions i and j, or how correlated are they (i.e. a value of zero indicates that the two dimensions are uncorrelated).

PCA: Goal Revisited

- Look for: $\mathbf{U}$
- S.t. : $[\mathbf{y}_1|\ldots|\mathbf{y}_N] = \mathbf{U}^T [\mathbf{x}_1|\ldots|\mathbf{x}_N]\ldots$

  & covariance is minimized

OR

- $\mathrm{Cov}(\mathbf{y})$ is diagonal
  - Note that $\mathrm{Cov}(\mathbf{y})$ can be expressed via $\mathrm{Cov}(\mathbf{x})$ and $\mathbf{U}$ as : $\mathrm{Cov}(\mathbf{y}) = \mathbf{U}^T \mathrm{Cov}(\mathbf{x})\, \mathbf{U}$

Selecting the Optimal U

How do we find such U ?

$$\lambda_i \mathbf{u}_i = \mathrm{Cov}(\mathbf{X})\mathbf{u}_i$$

Therefore :

Choose $\mathbf{U}_{opt}$ to be the eigenvectors matrix:

$$\mathbf{U}_{opt} = [\ \mathbf{u}_1 | \ldots | \mathbf{u}_d\ ]$$

where $\{\mathbf{u}_i | i=1,\ldots,d\}$ is the set of the d-dimensional eigenvectors of $\mathrm{Cov}(\mathbf{X})$!

So...to sum up

- To find a more convenient coordinate system one needs to :

Calculate mean sample μ → Subtract it from all samples x_i → Calculate Covariance matrix for resulting samples → Find the set of eigenvectors for the covariance matrix

↓

Create $\mathbf{U}_{opt}$, the projection matrix, by taking as columns the eigenvectors calculated !

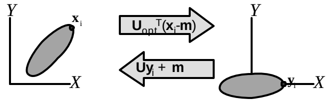
So…to sum up (cont.)

- Now we have that any point x_i can be projected to an appropriate point y_i by :

$$\mathbf{y}_i = \mathbf{U}_{opt}^T(\mathbf{x}_i - \mathrm{m})$$

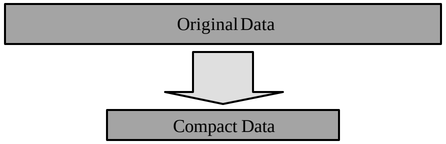
- and conversely (since $\mathbf{U}^{-1} = \mathbf{U}^T$)

$$\mathbf{U}\mathbf{y}_i + \mathrm{m} = \mathbf{x}_i$$

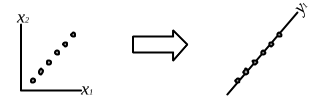


Data Reduction Using PCA

Reduce space dimensionality with minimum loss of description information.



Data Reduction: Example of an Ideal Case



2D data 1D data

Since there is no variance along one dimension, we only need a single dimension !!!

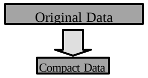
Data Reduction: Theory

- Each eigenvalue represents the the total variance in its dimension.
- Throwing away the least significant eigenvectors in $\mathbf{U}_{opt}$ means throwing away the least significant variance information !

Data Reduction: Practice

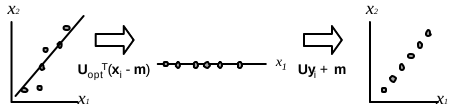
- Sort the d columns of the projection matrix $\mathbf{U}_{opt}$ in descending order of appropriate eigenvalues.
- Select the first m columns thus creating a new projection matrix of dimension d×m

This will now be a projection from a d-dimensional space to an m-dimentional space (m < d) !



Data Loss

- Sample points can still be projected via the new m×d projection matrix $\mathbf{U}_{opt}$ and can still be reconstructed, but some information will be lost.



2D data 1D data 2D data

PCA : Conclusion

- A multi -variant analysis method.
- Finds a new coordinate system for the sample data.
- Allows for data to be removed with minimum loss in reconstruction ability.

Eigenfaces

Face Recognition Problem

- Definition:
  - Given a database of labeled facial images
  - Recognize an individual in an unlabeled image formed from new and varying conditions (pose, expression, lighting etc.)
- Sub-Problems:
  - Representation:
    - How do we represent individuals?
    - What information do we store?
  - Classification:
    - How do we compare new data to stored information?

Representation

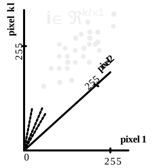
- Goal:
- Compact, descriptive object representation for recognition
- Representations:
  - Model - based Representation
    - Shape, texture, …
  - Appearance Based Representation
    - Images

Appearance Based Recognition

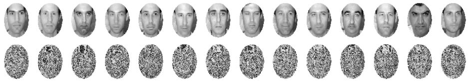
- Recognition of 3D objects directly from their appearance in ordinary images
- PCA / Eigenfaces:
  - Sirovich & Kirby 1987
    "Low Dimensional Procedure for the Characterization of Human Faces"
  - Turk & Pentland 1991
    "Face Recognition Using Eigenfaces"
  - Murase & Nayar 1995
    "Visual learning and recognition of 3D objects from appearance"

Images

$I \in \Re^{k \times l}$

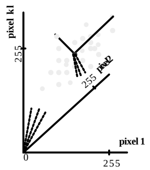


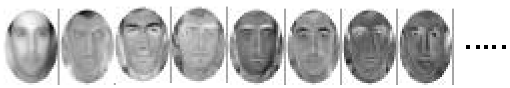
- An image is a point in $\Re^{kl \times 1}$ dimensional space



Eigenimages

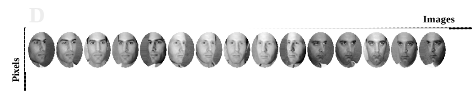
- Principal components (eigenvectors) of image ensemble





- Eigenvectors are typically computed using the Singular Value Decomposition (SVD) algorithm

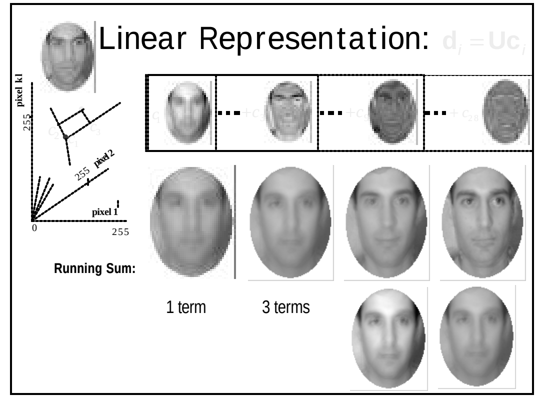
Matrix Decomposition - SVD



- A matrix has a column space and a row space
- SVD orthogonalizes these spaces and decomposes
- $D = U_1 S U_2^T$ (U_1 contains the eigenfaces)
- Rewrite in terms of *mode-n products*:

$$\boxed{D = S \times_1 U_1 \times_2 U_2}$$

Linear Representation: $d_i = Uc_i$



The Problem with Linear (PCA) Appearance Based Recognition Methods

- Eigenimages work best for recognition when only a single factor – e.g., object identity – is allowed to vary
- However, natural images are the consequences of ***multiple factors*** (or modes) related to scene structure, illumination and imaging

Perspective on Our Face Recognition Approach

	Linear Models	Our Nonlinear (Multilinear) Models
2^{nd} -Order Statistics (covariance)	**PCA** Eigenfaces	**Multilinear PCA** TensorFaces
Higher -Order Statistics	**ICA**	**Multilinear ICA** Independent TensorFaces

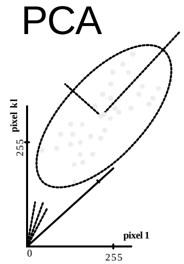
Vasilescu & Terzopoulos, CVPR 2005

ICA

Assumptions

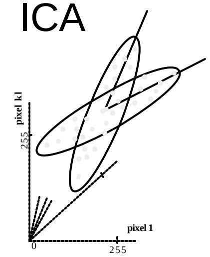
- The relationship between explanatory and response variable is linear
- Data has a non-gaussian distribution or at most one of the variables has a gaussian distribution

PCA



$$D = \underbrace{U}_{\text{basis matrix}} \underbrace{SV^T}_{\text{coefficient matrix}}$$

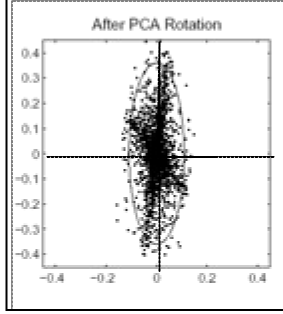
ICA



$$D = U \underbrace{W^T}_{} \underbrace{W^{-T}\ SV^T}_{}$$

$$= \underbrace{K}_{\text{independent components}}\ \underbrace{C}_{\text{coefficient matrix}}$$

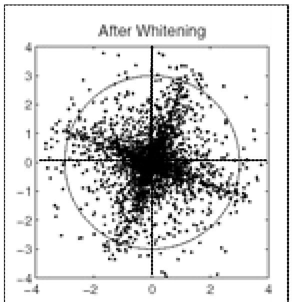
Geometric View of ICA



$$D = USV^T$$

$$D' = U^T D$$

Geometric View of ICA

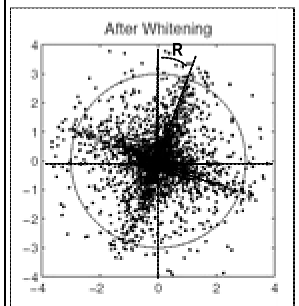


$$D = USV^T$$

$$D' = U^T D$$

$$D'' = S^{-\frac{1}{2}} U^T D$$

Geometric View of ICA



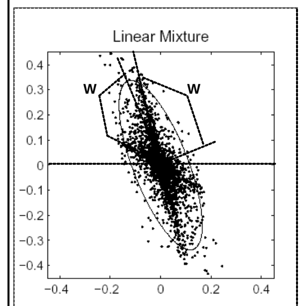
$$D = USV^T$$

$$D' = U^T D$$

$$D'' = S^{-\frac{1}{2}} U^T D$$

$$D''' = RS^{-\frac{1}{2}} U^T D$$

Geometric View of ICA



$$D = USV^T$$

$$D' = U^T D$$

$$D'' = S^{-\frac{1}{2}} U^T D$$

$$D''' = RS^{-\frac{1}{2}} U^T D$$

$$D = \underbrace{US^{\frac{1}{2}} R^T}_{} \underbrace{RS^{-\frac{1}{2}} SV^T}_{}$$

$$D = \underbrace{UW^{-1}}_{\text{Independent Components}} WSV^T$$

Independent Components