Anàlisi d'Imatges i Reconeixement de Formes
Image Analysis and Pattern Recognition:

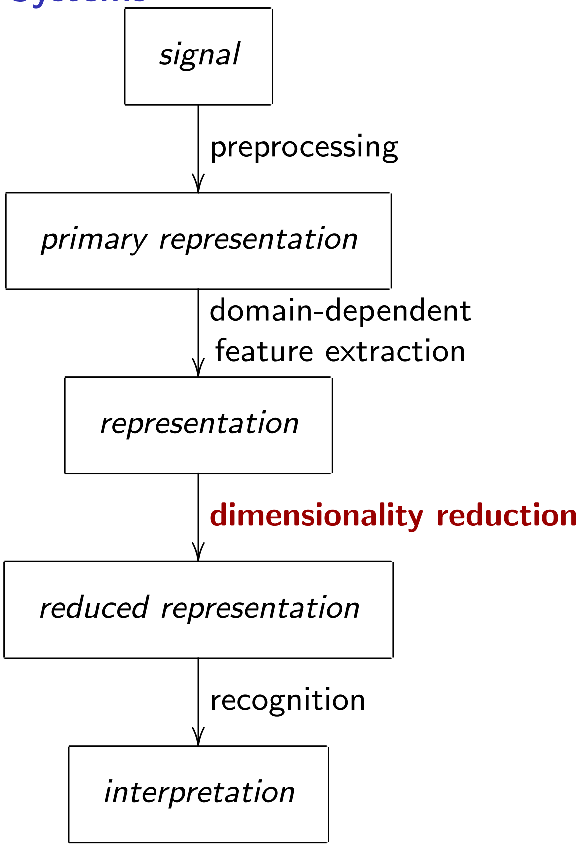
5. Complexity Reduction

Francesc J. Ferri

Dept. d'Informàtica. Universitat de València

Febrer 2008

Pattern Recognition Systems



signal

preprocessing

primary representation

domain-dependent feature extraction

representation

dimensionality reduction

reduced representation

recognition

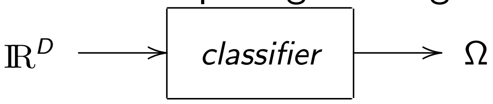
interpretation

Representation Spaces

- Usually several changes of representation (and corresponding spaces) are needed.
- We concentrate here in the domain-independent part of the process were no explicit apriori knowledge is used.
- Usually, there is no change of domain and transforms are restricted to **subspaces** or **manifolds** of the initial representation space.

Dimensionality reduction

The process can be seen as decomposing the original problem

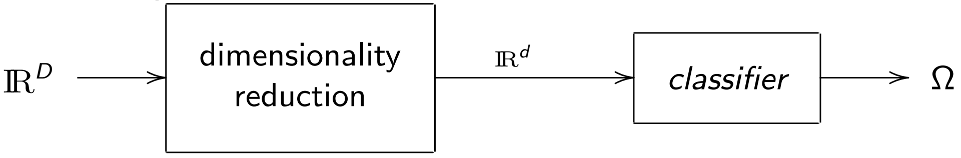


$\mathbb{R}^D$

classifier

Ω

into two different problems:



$\mathbb{R}^D$

dimensionality reduction

$\mathbb{R}^d$

classifier

Ω

in such a way that new learning problem (in a d-dimensional space) is significantly easier.

This is usually related to the fact that $d << D$.

Motivation

There is a number of reasons to consider dimensionality reduction:

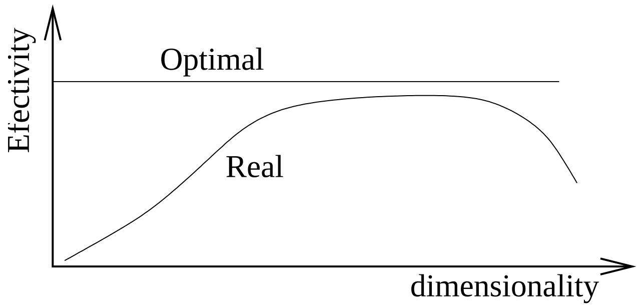
- Some of the attributes (dimensions) may contain no information or just noise.
- Large dimensionalities may lead to numerical instabilities when learning or estimating parameters.
- Too much dimensions obviously imply more computational cost and more hardware/space/transmission demands.
- Some of the measurements may imply an economic cost (medical diagnosis), which can be avoided if there is no significant impact of these in the learning/classification results.
- Obviously, there is a trade-off between the number of different measurements (dimensions) about a particular phenomenon and how particular learning methods make use of these to solve the classification problem.

The peaking phenomenon

Theoretically, in the asymptotic case, the more dimensions the better (even if just noise is added).

On the other hand, taking into account that we are usually interested in inductive learning where there is a process that depends on a training set of a particular size.

Consequently, there is a value over which adding new measures (dimensions) progressively degrades the results of any learning process based on a fixed size training set.



Feature extraction/selection

Feature Extraction

$$\begin{tikzcd} (y_1, \ldots, y_D) \arrow[r] & \boxed{A(y_1, \ldots, y_D)} \arrow[r, "(x_1, \ldots, x_d)"] & \boxed{\mathcal{C}} \arrow[r] & \Omega \end{tikzcd}$$

↑ Potentially optimal
↓ Must restrict the family of functions

↑ Less information is lost
↓ the function must be used online also

Feature Selection

$$\begin{tikzcd} (y_1, \ldots, y_D) \arrow[r] & \boxed{j(i)} \arrow[r, "(y_{j_1}, \ldots, y_{j_d})"] & \boxed{\mathcal{C}} \arrow[r] & \Omega \end{tikzcd}$$

j_i represents any particular subset of the set of features $\mathcal{U} = \{1, \ldots, D\}$.

↑ Discarded features are not needed
↓ More information is lost

↑ Selected features keep their meanings
↓ Combinatorial problem (usually NP-complete)

Formulation

The goal of feature extraction is to find a mapping:

$$W = \arg \max_{w \in \mathcal{W}} J(w)$$

The goal of feature selection is to find a subset:

$$\chi = \arg \max_{\xi \in \mathcal{U}} J(\xi)$$

The criterion J is suposed to measure the degree of efectivity that can be achieved by using either the transformed or selected features when solving the classification problem.

Feature extraction/selection criteria

- Probability of error
- Probability distances

Battacharyya distance

$$J_B = -\ln \int \sqrt{p(x'|w_1)p(x'|w_2)dx'}$$

Kullback-Leibler divergence

$$J_K = \int p(x'|w_1) \ln \frac{p(x'|w_1)}{p(x'|w_2)} dx'$$

- Entropy measures
- Dependence measures
- Inter- intra- class distance measures

Feature extraction/selection criteria

The Battacharyya distance in the normal case is

$$J_B = \frac{1}{4}(\mu_2 - \mu_1)^T [\Sigma_1 + \Sigma_2]^{-1} (\mu_2 - \mu_1) + \frac{1}{2} \ln \left[\frac{\frac{1}{2}(\Sigma_1 + \Sigma_2)}{\sqrt{|\Sigma_1||\Sigma_2|}} \right]$$

In the particular case of equal covariance matrices this reduces to the Mahalanobis distance

$$J_M = (\mu_2 - \mu_1)^T \Sigma^{-1} (\mu_2 - \mu_1)$$

The Feature Selection Problem

In the most general case, the only optimal way of solvint the FS problem is though **exhaustive exploration**.

The search space can be of size $\binom{D}{d}$ (subsets of size d) or 2^D (all subsets of $\mathcal{U}$).

For each subset, the criterion J must be computed. Apart from the search problem, the computation of J may lead to significantly different methods.

Monotonic criteria

A function $J : 2^{\mathcal{U}} \rightarrow \mathbb{R}$ is said to be **monotonic** if

$$\chi_i \subset \chi_j \Rightarrow J(\chi_i) \leq J(\chi_j), \quad \forall \chi_i, \chi_j \subseteq \mathcal{U}$$

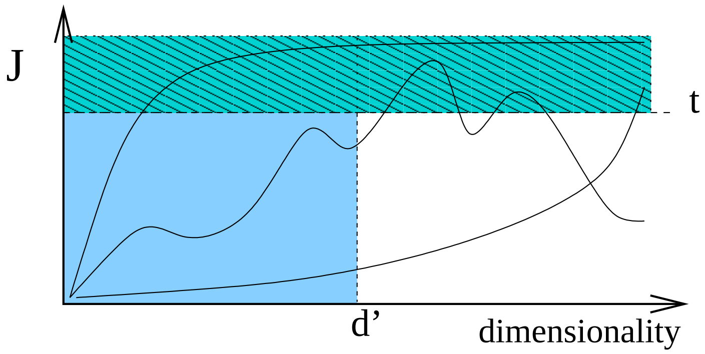
Most probabilistic distances, entropy and dependence measures are monotonic.

Optimization

Once the criterion has been selected, the problem can be formulated in two different ways.

$$\begin{array}{l} \text{Minimize } d \text{ subject to } J(\chi_d) > t \text{ (given } t) \\ \text{Maximize } J(\chi_d) \text{ subject to } d < d' \text{ or } d = d') \text{ (given } d') \end{array}$$

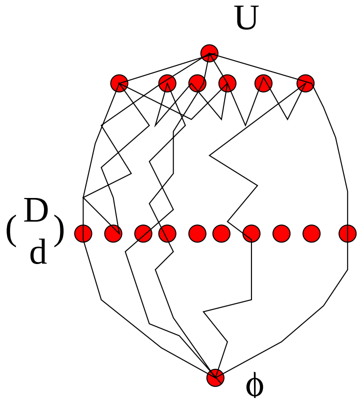
Some methods give as a result a sequence of subjects of all dimensionalities.

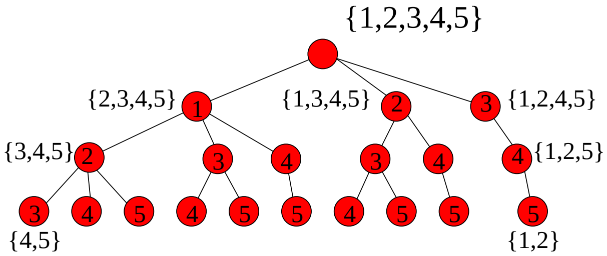


Optimal feature selection

If J is monotonic, the best d-subset of features can be obtained in a slightly more efficient way.

The feasible part of the search space are the $\binom{D}{d}$ subsets. It is possible to conveniently enumerate this

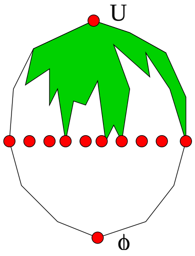




Optimal feature selection: bounding

As J is monotonic and all subsets along any of the branches of the trees are nested, one can stop exploring a subtree as soon as a value worse than the best subset so far.

This search can be improved if features are ordered at each node according to partial values of the criterion computed.



This is known as the branch and bound algorithm for feature selection.

Bad news: exponential worst case!!

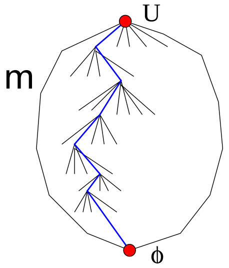
Sequential Feature Subset Search

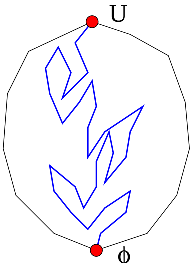
It is a whole family of sequential algorithms that explore the lattice of subsets in a simple fashion either top-down (Backward) or upwards (Forward).

Sequential Forward Selection (SFS)

Sequential Backward Selection (SBS)

Plus ℓ, take away r





Sequential Feature Subset Search

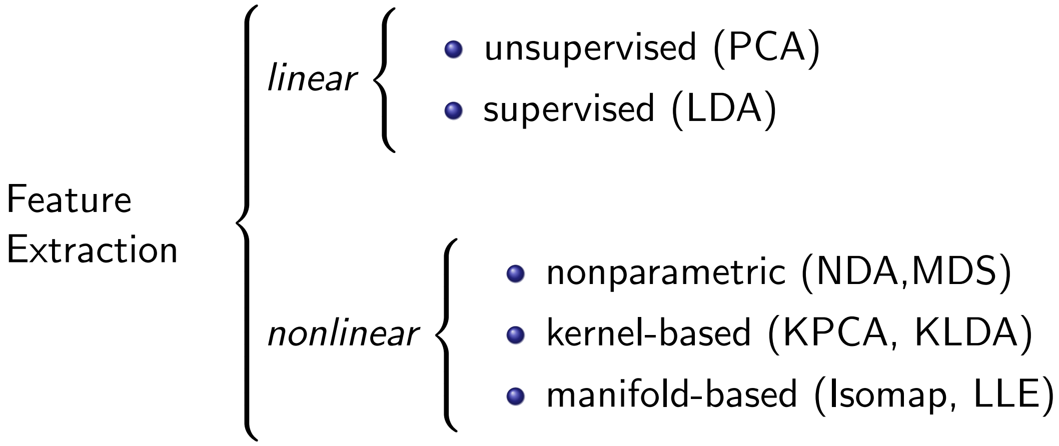
Generalized plus ℓ take away r

Sequential Floating Forward Selection (SFFS)

Other approaches for FS

-Use metaheuristic search to explore the search space.
-Use mixtures of gaussians and slack variables to select features
-Use boosting algorithms to learn weights to rank features
-etc.

Taxonomy



- Feature Extraction
  - *linear*
    - unsupervised (PCA)
    - supervised (LDA)
  - *nonlinear*
    - nonparametric (NDA,MDS)
    - kernel-based (KPCA, KLDA)
    - manifold-based (Isomap, LLE)

Unsupervised Feature Extraction

The easiest case of dimensionality reduction consists of looking for a linear transformation of the original data

$$\mathbf{y} = A^T \mathbf{x}$$

$\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{y} \in \mathbb{R}^d$ and A is an $D \times d$ matrix.

A set of n column vectors X can be transformed as

$$(\mathbf{y_1} \cdots \mathbf{y_n}) = Y = A^T X = A^T (\mathbf{x_1} \cdots \mathbf{x_n})$$

Principal Component Analysis

The goal is to find a linear transformation, $y = A^T x$, that gives optimally uncorrelated data. That is, $E[y_i y_j] = 0 \ \forall i \neq j$.

Equivalently, the **correlation matrix**, $R_x = E[xx^T]$ should be diagonal.

Two almost equivalent matrices also used in this context are the **covariance matrix**, $\Sigma_x = E[(x-\mu)(x-\mu)^T]$, and the **scatter matrix**, $S_x = \sum_k (x_k - \mu)(x_k - \mu)^T$.

The transform A is obtained by eigenanalysis of any of the above matrices.

$$A = (\mathbf{u_1}, \ldots, \mathbf{u_D})$$

$$S_x \mathbf{u_k} = \lambda_k \mathbf{u_k}$$

$$A^T S_x A = \Lambda$$

where Λ is a diagonal matrix with all eigenvalues, λ_k.

Principal Component Analysis

If less than D eigenvectors are used, we have a $D \times d$ matrix and the transformation cannot be exactly inverted.

$\hat{\mathbf{x}} = A\mathbf{x}$ is the reconstruction of $\mathbf{x}$. It can be shown that the reconstruction error is given by

$$E[||\mathbf{x} - \hat{\mathbf{x}}||^2] = \sum_{discarded} \lambda_k$$

This transform is usually applied with a target dimension d or a fixed percentage of variance preserved.

This is referred to also as Karhunen-Loeve, Hotelling, or Whitening.

The Fisher mapping

Let $S_W = \sum_i^c S_i$ be the within-class scatter matrix (S_i is the i-th class scatter matrix).

And let $S_B = \sum_i^c n_i(\mu_i - \mu)(\mu_i - \mu)^T$ the between-class scatter matrix

$(S_T = S_W + S_B)$

The generalized Fisher criterion is

$$J(A) = \frac{|A^T S_B A|}{|A^T S_W A|}$$

which gives rise to a generalized eigenvalue problem $S_B\mathbf{w} = \lambda S_W\mathbf{w}$.

The solution is given by the eigenvectors of $S_W^{-1}S_B$ (if S_W is invertible) $(c-1$ at most).