TÉCNICO+
FORMAÇÃO AVANÇADA

Dimensionality reduction

Feature selection, principal component analysis, discriminant analysis

DASH: Data Science e Análise Não Supervisionada

Rui Henriques, rmch@tecnico.ulisboa.pt
Instituto Superior Técnico, Universidade de Lisboa

Outline

- Dimensionality reduction: why and how
- Feature selection
- Linear transformations
  - algebra ground
  - principal component analysis
  - compression and reconstruction
  - non-linear kernels
  - linear discriminant analysis
- Subspace selection
- Evaluation
- Data reduction

Motivation

- At a first glimpse, increasing the number of variables should improve learning…
- In practice, including more variables can degrade performance (i.e. **curse of dimensionality**)
  - *challenges*: learning complexity and generalization difficulty (over/underfitting)
  - common definition of **high-dimensionality**: $|Y| \gg |X|$ (i.e. $m \gg n$)
- The number of training observations required increases **exponentially** with dimensionality
- How then can we learn in high-dimensional data spaces with a limited number of observations?
  - **dimensionality reduction**

Data domains with high-dimensionality

- **text** and **web content** data (*left*)
- **social** behavioral data
- **biological data**
  - genetic variants (millions per individual)
  - gene expression (>20k genes)
  - molecular concentrations (metabolites, proteins…)
- **healthcare** data (clinical records)
- **consumer data**
- **signal, audio, image** and **video** data

thousands of terms

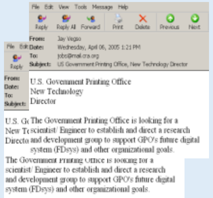
documents

	T_1	T_2	………	T_m	c
D_1	12	0	………	6	sports
D_2	3	10	………	28	travel
⋮	⋮			⋮	⋮
D_n	0	11	………	16	jobs

Task: classify unlabeled documents
Challenge: thousands of terms
Solution: dimensionality reduction

web pages

emails

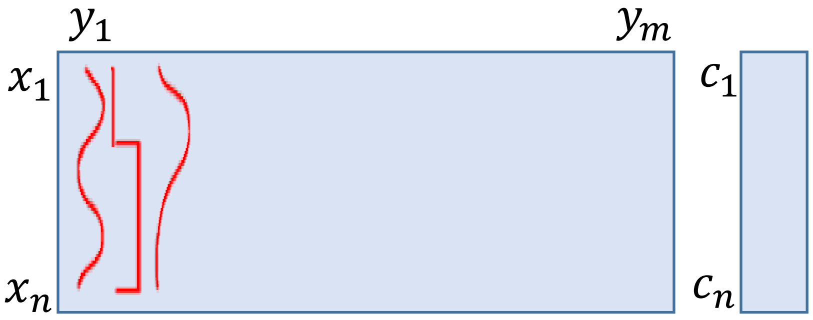


Challenges

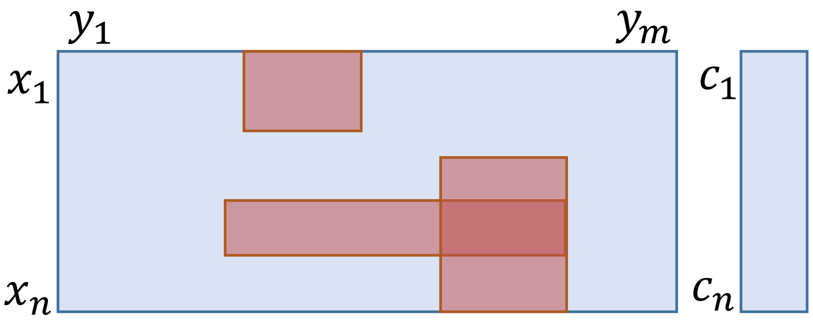
High-dimensional data analysis:

- learning **complexity**: large data space
- **generalization difficulty** (insufficient observations)
  - **overfitting** risk
  - **underfitting** risk
- how to evaluate these risks?
  - behavior for varying data size
  - compare training and testing error
  - variability of errors across testing folds
  - bias and variance components of error

Generalization: overfitting and underfitting risks



Overfitting
- unability to discard non-informative and/or non-discriminative data



Underfitting
- exclusion of informative or discriminative data from the learning

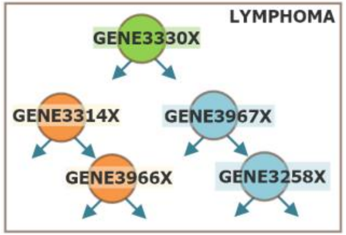
Global and local learning

"Global" learning

- descriptors (e.g., *clusters*) and predictors (e.g., *discriminants*)
  - all features considered, often equally relevant in joint probabilistic stances (e.g. *naïve Bayes*)
- Problem? **overfitting**
  - what if only a few variables are relevant?

"Local" learning

- descriptors (*patterns*) and predictors (e.g. *decision trees, kNN*)
  - few combined features or observations as long as they are informative or discriminative
- Problem? **underfitting**
  - many potentially relevant features or observations neglected



Dimensionality reduction

Some **goals** of dimensionality reduction:

- Guide **supervised learning** (focus on discriminative regions)
- Guide **unsupervised learning** (focus on informative regions)
- **Visualization** (project high-dim data into interpretable low-dim data)
- **Data compression** (efficient storage and retrieval)
- **Noise removal** (denoising data)
- **Speed-up** learning
- **Describe** the underlying properties of data
- Guarantee simplicity and comprehensibility of mined results
- Map **multimedia data** (image and signal data) into feature-based data
- Support matrix operations (inverse, rank determination, approximation)…

How?

1. **Feature selection**
2. **Feature extraction**/transformation
    - principal component analysis
    - linear discriminant analysis
    - representation learning
3. **Sparse kernels** and **regularization** in parametric models to exclude non-relevant parameters
    - neural networks, support vector machines, discriminant analysis…
4. **Subspace selection** to jointly select variables and observations
    - pattern mining, decision trees and random forests
    - associative classifiers and decision tables

Dimensionality reduction

- Project the m-dimensional observations into a k-dimensional space ($k \ll m$)
  - preserve most of relevant information or structure from data
- Solve the learning problem in low dimensions
- Two major approaches
  - **feature selection**
    - choosing a subset of all features

$$[y_1, y_2, .., y_m] \longrightarrow [y_{i1}, y_{i2}, .., y_{ik}]$$

  - **feature extraction**
    - creating new features by combining existing ones

$$[y_1, y_2, .., y_m] \longrightarrow [d_1, d_2, .., d_k] \text{ using } f([y_{i1}, y_{i2}, .., y_{im}])$$

Outline

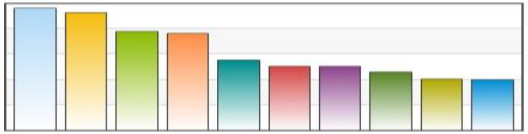
- Dimensionality reduction: why and how
- **Feature selection: recall**
- Linear transformations
  - algebra ground
  - principal component analysis
  - compression and reconstruction
  - non-linear kernels
  - linear discriminant analysis
- Subspace selection
- Evaluation
- Data reduction

Feature selection

- Optimal subset of features according to an objective function
  - differs from feature extraction where output features are (non-)linear combinations of original features
- **Objective criteria**
  - **unsupervised** setting: *variables with higher variability and entropy*
  - **supervised** setting: *maximize discrimination/correlation with target variable*
- **Perspectives**
  - subset search vs. feature ranking
  - models: filter vs. wrapper

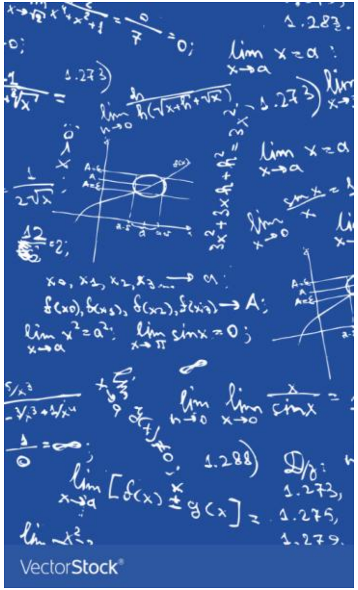
Feature ranking

- Weighting each individual feature according to objective criteria (e.g. information gain)



- Sort and select **top-ranked features**: a) threshold, b) *k*-top, or c) percentile
- Disadvantages
  - hard to determine threshold or *k*
  - unable to consider correlation between features
- Advantages
  - efficient $O(m)$ no need to test combination of features (subset optimality)

Measures for feature ranking



Goodness of a feature/feature subset:

- **information measures**
- **correlation/association measures**
- **distance measures**
- **accuracy measures**

Information measures in classification

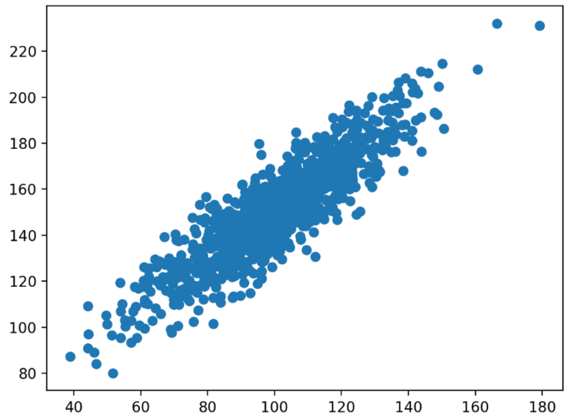
- **χ^2 test** (`chi2` in Python)
  - robust for input categorical variables
  - two values: χ^2 statistic (higher the better), *p*-value

```
iris = datasets.load_iris()
X, y = iris.data, iris.target

chi2, pval = chi2(X, y)

[ 10.81782088 3.59449902 116.16984746 67.24482759]
[ 4.47651e-03 1.6575e-01 5.943443e-26 2.50017e-15]
```

- **ANOVA test** (`f_classif` in Python)
  - robust for numeric input variables
  - also valid for categorical input with numerical output
  - two values: F-value statistic (higher the better), *p*-value

Information measures in regression

- **correlation** for numeric input variables:
  - Pearson and Spearman (see previous lectures)
  - F-statistic (higher the better) and accompanying p-value



- **mutual information** or **ANOVA**
  for categorical input variables

- want to abstract the type of input variables?
  use `f_regression` from sklearn

Information measures in description

- as a ***filter***: measure feature importance and select top-*k* features or above threshold
  - **unsupervised** setting: **high entropy**
  - supervised setting: next slides
- as a ***wrapper***: assess learning performance with varying subsets of features
  - simple: measure feature importance and test descriptors on top-*k* features with varying *k*
  - advanced: assess descriptors with different subsets of variables (irrespectively of top-*k*)
  - how to assess descriptors?
    - e.g. clustering quality – any problem when considering silhouette?

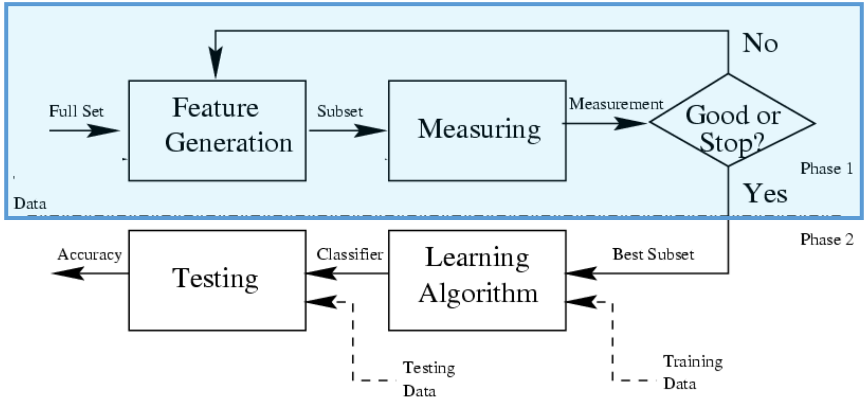
Feature selection: filter *vs.* wrapper

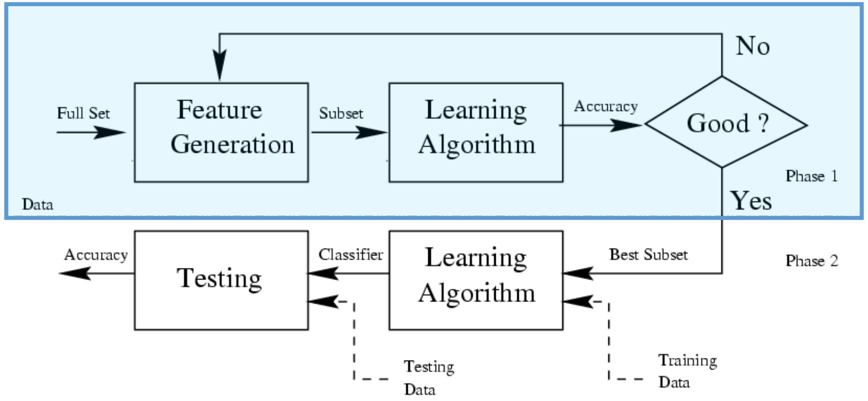
Filter model

- independent from the learning algorithm
    - efficient and no learning biases
- rely on general characteristics of data

Wrapper model

- evaluated on a given descriptor/predictor
- descriptive utility or predictive accuracy as a goodness measure
- better yet dependent on the learning approach
- more computationally expensive



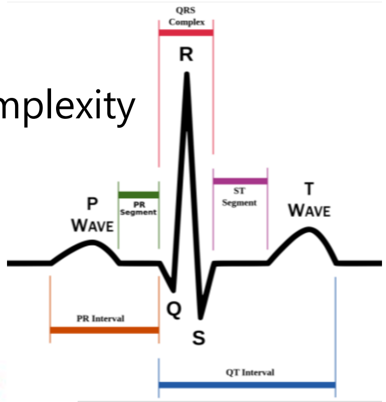


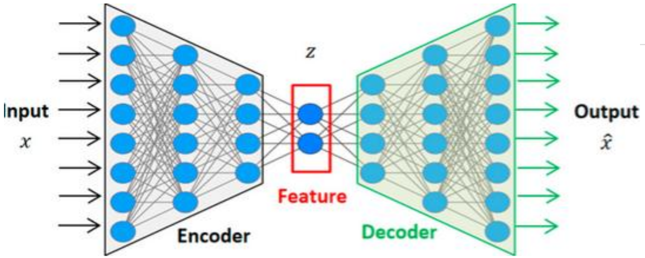
Outline

- Dimensionality reduction: why and how
- Feature selection
- **Linear transformations**
    - **algebra ground**
    - **principal component analysis**
    - compression and reconstruction
    - non-linear kernels
    - linear discriminant analysis
- Subspace selection
- Evaluation
- Data reduction

Feature extraction

- Feature extraction used to
  - **extract relevant features from complex data** to handle structural complexity
    - extract domain-specific features of interest
      - e.g. PQRST statistics from ECG signal
    - extract domain-independent statistics of interest
      - e.g. spectral statistics from time series and images
    - learn latent features using neural networks (**data representation**)





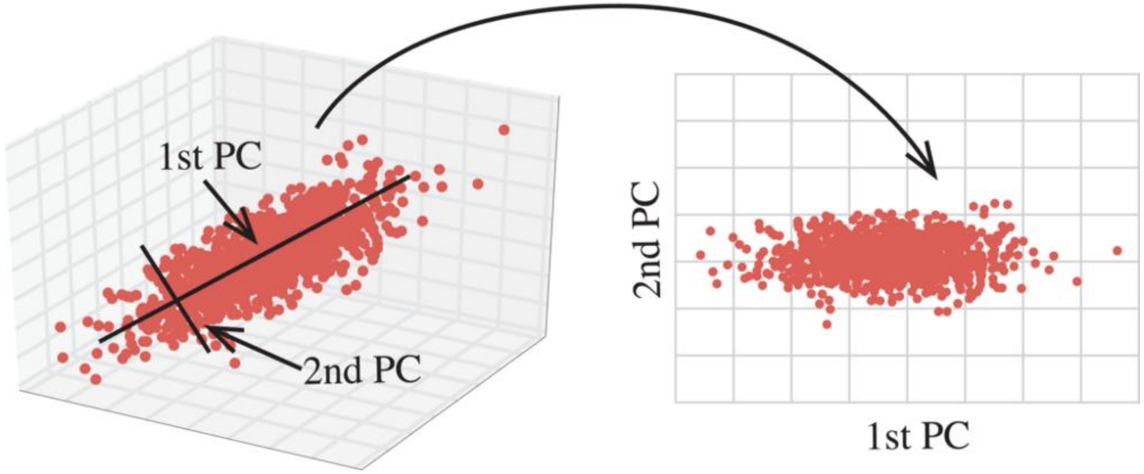
  - **learn transformations** that explain simple multivariate data with reduced dimensionality

Feature extraction

Feature extraction used to

- extract relevant features from complex data to handle structural complexity
- **learn transformations** that explain simple multivariate data with reduced dimensionality: **TODAY!**
  - classical approaches aim at finding a *linear transformation*
    - Why? simple to compute and analytically tractable
    - Goal: reduction that preserves as much information in data as possible
      - paradigmatic case: **Principal Component Analysis** (PCA)
  - simple extensions available
    - *non-linear transformations* (kernel trick)
    - accommodate discriminative power in *supervised settings*
      - Goal: reduction that best separates the data
        - paradigmatic case: **Linear Discriminant Analysis** (LDA)

Algebra ground for PCA



Axes of greater variance given by *eigenvectors* of *covariance matrix*

Algebra ground

- A **covariance matrix** measures the tendency of two variables to vary in the same direction
  - positive (negative) covariance denote positive (inverse) correlation
    - nevertheless magnitude of values not easily interpretable
    - when normalizing covariance by their variances we obtain linear correlation in [-1,1]
  - covariance matrix is symmetric and positive-definite
- Remember
  - sample covariance: $n-1$ in the denominator (Bessel's correction)

$$cov(y_1, y_2) = \frac{\sum_{i=1}^{n}(a_{1i} - \overline{y_1}) \cdot (a_{2i} - \overline{y_2})}{n-1}$$

  - whole population: n is the denominator $$cov(y_1, y_2) = \frac{\sum_{i=1}^{n}(a_{1i} - \overline{y_1}) \cdot (a_{2i} - \overline{y_2})}{n}$$

TÉCNICO+
FORMAÇÃO AVANÇADA

Algebra ground

- **Covariance matrix**, given m attributes, $y_1, \ldots, y_m$

$$C^{m \times m} = (c_{ij} \mid \text{i,j} = 1..\text{m}), \text{ where } c_{ij} = \text{cov}(y_i, y_j)$$

	Hours(H)	*Mark(M)*
Data	9	39
	15	56
	25	93
	14	61
	10	50
	18	75
	0	32
	16	85
	5	42
	19	70
	16	66
	20	80
Totals	167	749
Averages	13.92	62.42

$$C^{m \times m} = \begin{pmatrix} \text{cov}(H,H) & \text{cov}(H,M) \\ \text{cov}(M,H) & \text{cov}(M,M) \end{pmatrix}$$

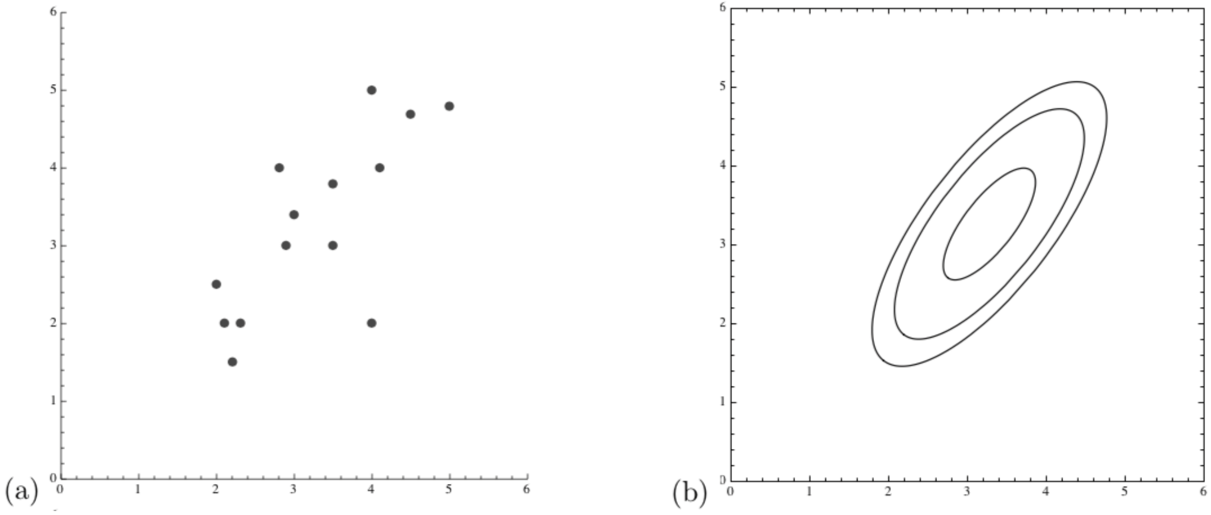
$$= \begin{pmatrix} \text{var}(H) & 104.5 \\ 104.5 & \text{var}(M) \end{pmatrix}$$

$$= \begin{pmatrix} 47.7 & 104.5 \\ 104.5 & 370 \end{pmatrix}$$

TÉCNICO+
FORMAÇÃO AVANÇADA

Algebra ground

- The covariance matrix of the data points defines the ellipses of equiprobability on the right



Eigenvalues and eigenvectors

- Let C be a $m \times m$ covariance matrix
- Vectors $\mathbf{v}$ having same direction as $C\mathbf{v}$ are called *eigenvectors*
  - eigenvectors define the linear composition of attributes
- In the equation $\boxed{C\mathbf{v} = \lambda\mathbf{v}}$, λ is called an *eigenvalue* of C
- Example:

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4\begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$\mathbf{v} = [3\ 2]^T$ and $\lambda = 4$

meaning that data is described by $y_{new} = 3y_1 + 2y_2$

Eigenvalues and eigenvectors

- $A\mathbf{v} = \lambda\mathbf{v} \Leftrightarrow (A - \lambda I)\mathbf{v} = 0$
- Given A, how to calculate $\mathbf{v}$ and λ:
  - determine roots to $det(A - \lambda I) = 0$, roots are eigenvalues λ
  - solve $(A - \lambda I)\mathbf{v} = 0$ for each λ to obtain eigenvectors $\mathbf{v}$

y_1	y_2
-5.1	9.25
14.9	20.25
5.9	33.25
5.9	-30.75
…	…
-9.1	-10.75
-9.1	-21.75
5.9	19.25

$$C = \begin{pmatrix} 2 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}$$

Eigenvectors and eigenvalues:

$\mathbf{v}_1 = [0.91, 0.41], \lambda_1 = 2.36$

$\mathbf{v}_2 = [-0.41, 0.91], \lambda_2 = 0.23$

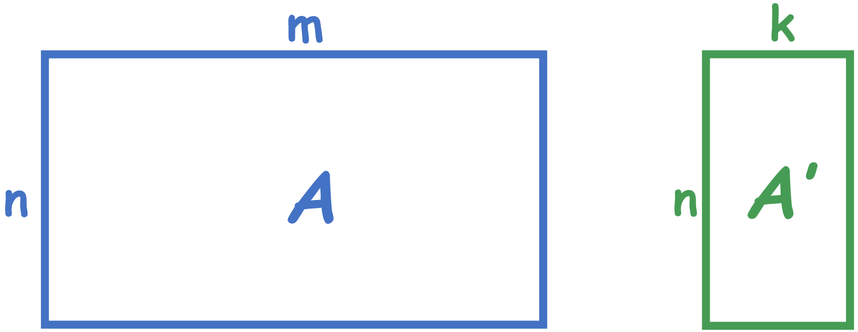
$$\mathbf{x}_i = (0.91 \quad 0.41) \begin{pmatrix} a_{i1} \\ a_{i2} \end{pmatrix}$$

y_{new}
-0.8
21.9
19
-7.2
…
-12.7
-17.2
13.3

TÉCNICO+
FORMAÇÃO AVANÇADA

Dimensionality reduction

- Map data with m variables into k variables (such that $k < m$) without significant loss

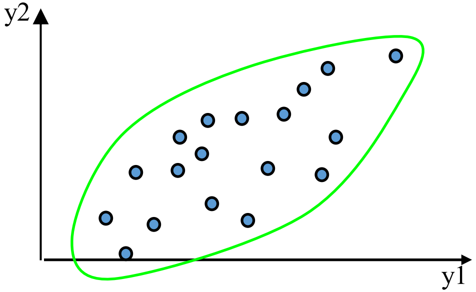


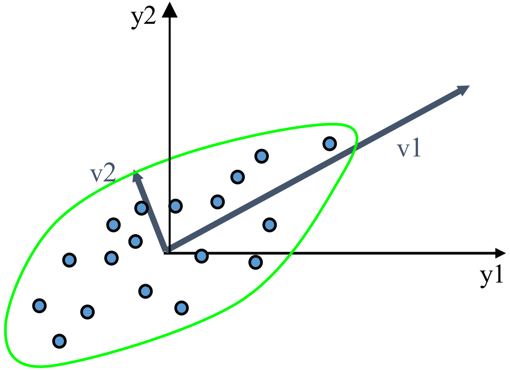
- *Residual variation*: information in A not retained in A′
- Trade-off: dimensionality (*k*) and interpretability *versus* information loss

Linear transformations

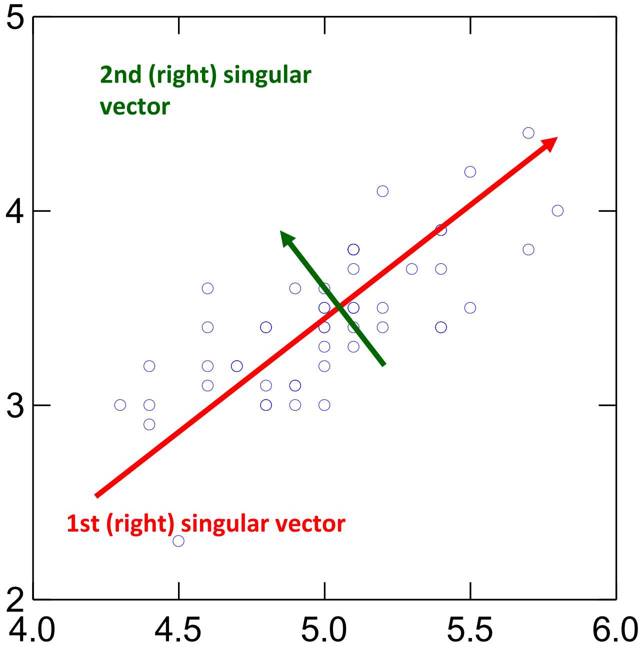
Intuition:
- find the axis that shows the greatest variation
- project all points into this axis

How:
- move origin to the center of the dataset
- find the eigenvectors and eigenvalues of the data covariance matrix
  - the eigenvectors define the new data space





Singular value decomposition (SVD)



1^{st} singular vector: direction of maximal variance
λ_1**:** how much data variance is explained by 1^{st} vector

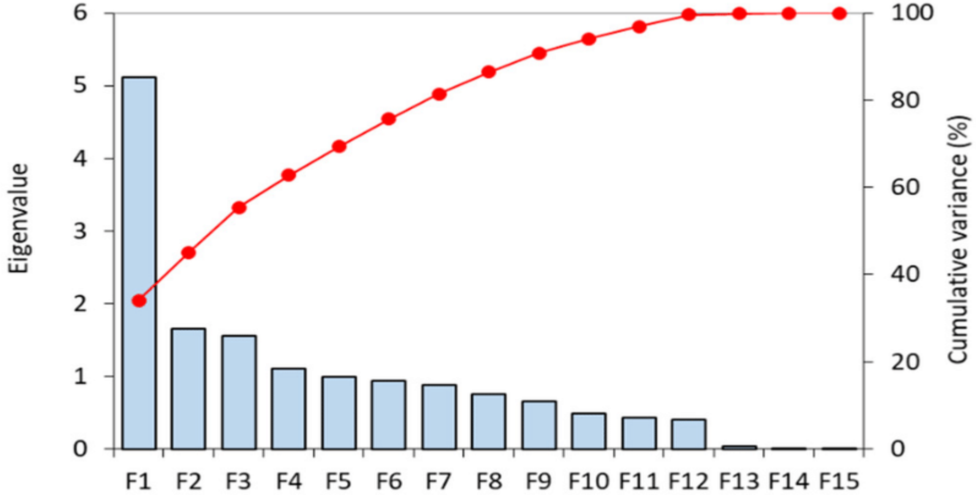
2^{nd} singular vector: direction of maximal variance, after removing projection of 1^{st} vector
λ_2**:** how much data variance is explained by the 2^{nd} vector

…

k^{th} **singular vector** …
(until unexplained variance below threshold)

Principal component analysis (PCA)

- PCA is SVD done on *centered* data
  - singular vector/value = eigenvector/value
- First component (PC1): highest eigenvalue (direction with greatest variation)
- Second component (PC2): direction with maximum variation orthogonal to PC1
- ...



TÉCNICO+
FORMAÇÃO AVANÇADA

Component selection

- The variance in the direction of the k^{th} eigenvector is given by the eigenvalue λ_k
- Singular values can be used to estimate how many components to keep
- **Rule of thumb:** keep enough to explain 85% of the variation

$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{m} \lambda_j} \approx 0.85$$

if $k = m$, we preserve 100% of the original variation

  - depending on the reduced dimensionality and learning needs: 90%, 95%, 98% also common
- **Karhunen-Loeve** (KL) transform is PCA without subsequent removal of components
  - no information loss

Principal component analysis

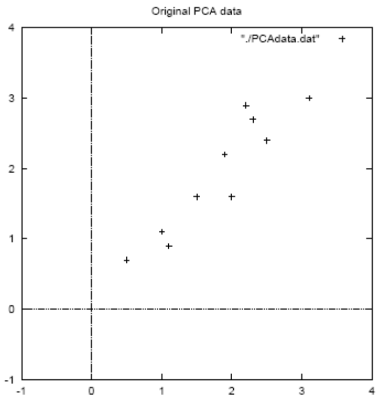
- PCA projects data along the directions where the data varies most
  - directions are determined by the eigenvectors with the largest eigenvalues
  - reduction can imply information loss, yet PCA preserves as much information as possible

- Components (summary variables)
  - linear combinations of the original variables
  - uncorrelated with each other
  - the largest eigenvalues are called ***principal components***
    - the eigenvalue is the magnitude of eigenvector, defining the direction's variance
    - in general: *only few components needed to capture most data variability*

Principal component analysis

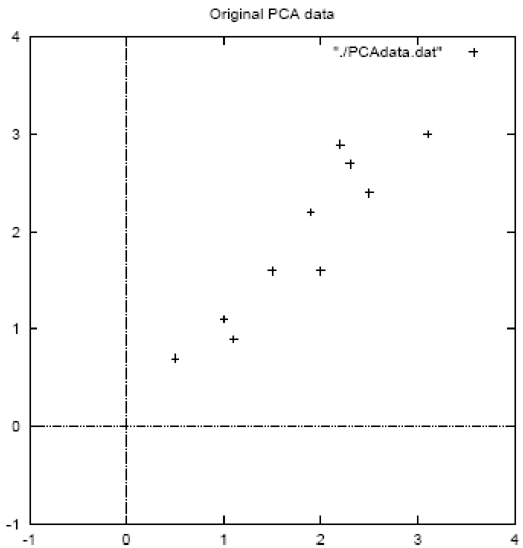
- Revising **how**
  1. Subtract mean of each variable
  2. Compute the covariance matrix $m \times m$ (*scatter* of data)
  3. Compute eigenvalues, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$, and eigenvectors, $v_1, v_2, \ldots v_m$
  4. Keep the large k eigenvalues ($k \leq m$) and construct the transformed space
  5. Transform the dataset $D \rightarrow D'$

- **Exercise**: apply PCA on the following dataset



PCA: Example

1. Centering data



OriginalData =

y_1	y_2
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

mean 1.81 | 1.91

CenteredData =

y_1	y_2
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

mean 0 | 0

TÉCNICO+
FORMAÇÃO AVANÇADA

PCA example

2. Calculate the covariance matrix:

$$
\begin{array}{ccc}
 & y_1 & y_2 & \\
cov = & \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix} & \begin{array}{c} y_1 \\ y_2 \end{array}
\end{array}
$$

3. Calculate its (unit) eigenvectors and eigenvalues

$$
eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix} \qquad eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}
$$

4. Order eigenvectors by eigenvalue, highest to lowest and select top *p*

$$
\mathbf{v}_1 = \begin{pmatrix} -.6779 \\ -.7352 \end{pmatrix} \quad \lambda_1 = 1.284 \qquad \mathbf{v}_2 = \begin{pmatrix} -.7352 \\ .6779 \end{pmatrix} \quad \lambda_2 = .0491
$$

… and construct the transformed feature vector

$$
FeatureVector(k=2) = \begin{pmatrix} -.6779 & -.7352 \\ -.7352 & .6779 \end{pmatrix} \qquad FeatureVector(k=1) = \begin{pmatrix} -.6779 \\ -.7352 \end{pmatrix}
$$

PCA example

5. Derive the new data set

   TransformedData = RowFeatureVector × RowDataAdjust

$$DataAdjusted = \begin{pmatrix} .69 & -1.31 & .39 & .09 & 1.29 & .49 & .19 & -.81 & -.31 & -.71 \\ .49 & -1.21 & .99 & .29 & 1.09 & .79 & -.31 & -.81 & -.31 & -1.01 \end{pmatrix}$$

$$FeatureVector(p=2)^T = \begin{pmatrix} -.6779 & -.7352 \\ -.7352 & .6779 \end{pmatrix}$$

$$FeatureVector(p=1)^T = \begin{pmatrix} -.6779 & -.7352 \end{pmatrix}$$

TransformedData (p=2) =

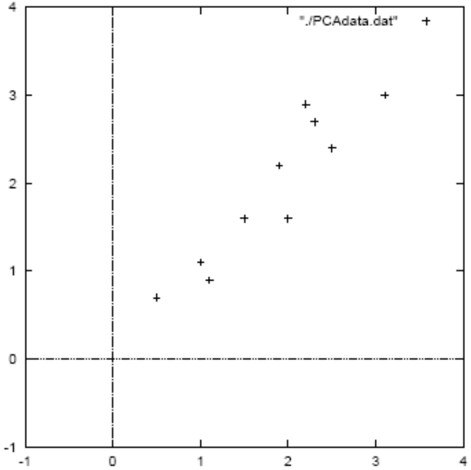
c_1	c_2
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

TransformedData (p=1) =

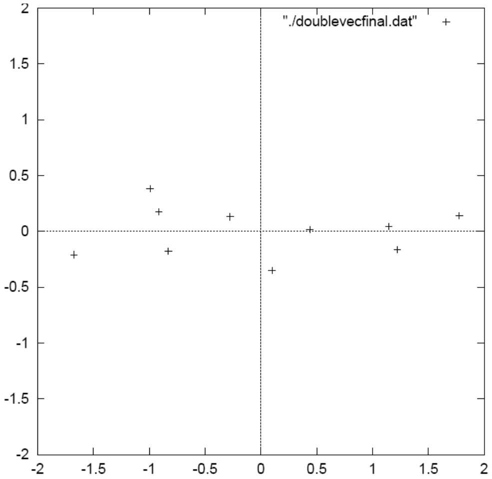
c_1
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

PCA example

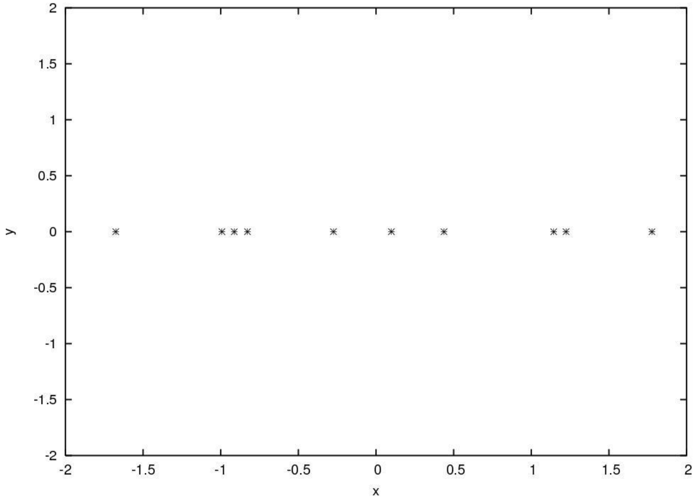
Original data



Data transformed with
$k = 2$ eigenvectors



Data transformed with
$k = 1$ eigenvector



Outline

- Dimensionality reduction: why and how
- Feature selection
- **Linear transformations**
  - algebra ground
  - principal component analysis
  - **compression and reconstruction**
  - non-linear kernels
  - linear discriminant analysis
- Subspace selection
- Evaluation
- Data reduction

Compression

- **Goal**: compress data without information loss
- Why? Remember…
    - **description**
        - compactness, interpretability, simplicity
        - transforms can be used to explain relevant information (variable dependencies)
        - how? Assess which variables are contributing more to each component
            - remember a component is a linear composition of variables
    - guide **prediction** on the compressed data space
    - **denoising** data
    - **efficient** data storage, retrieval, learning

Recovering original data

- The original vectors can be reconstructed using its principal components
- Given the reduced data space, how to retrieve/decompress old data?

$$DataRecovered = (FeatureVector \times TransformedData) + OriginalMean$$

we did: $\underline{TransformedData = FeatureVector^{T} \times DataAdjust^{T}}$

so we can do: $\underline{DataAdjust^{T}} = (FeatureVector^{T})^{-1} \times TransformedData$

(orthogonal property) $\underline{= FeatureVector \times TransformedData}$

and: $\underline{DataRecovered = DataAdjust + OriginalMean}$

Reconstruction: example

$$FeatureVector(p=2)^T = \begin{pmatrix} -.6779 & -.7352 \\ -.7352 & .6779 \end{pmatrix}$$

$$FeatureVector(p=1)^T = \begin{pmatrix} -.6779 & -.7352 \end{pmatrix}$$

y_1	y_2
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

c_1	c_2
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

using both components

y'_1	y'_2
0.69	0.49
-1.31	-1.21
0.39	0.99
0.09	0.29
1.29	1.09
0.49	0.79
0.19	-0.31
-0.81	-0.81
-0.31	-0.31
-0.71	-1.01

after uncentering

y'_1	y'_2
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

DataRecovered = (*FeatureVector* x *TransformedData*) + *OriginalMean*

Reconstruction error

- PCA minimizes the reconstruction error: $\|\mathbf{x} - \hat{\mathbf{x}}\|$
- It can be shown that the reconstruction error is: $error = 1/2 \sum_{i=k+1}^{m} \lambda_i$
  - using 2 components: recovery error = 0 (from previous slide)
  - using 1 component

y'_1	y'_2	y^*_1	y^*_2
0.56	0.61	2.4	2.5
-1.20	-1.31	0.6	0.6
0.67	0.73	2.5	2.6
0.19	0.20	2.0	2.1
1.14	1.23	2.9	3.1
0.62	0.67	2.4	2.6
-0.07	-0.07	1.7	1.8
-0.78	-0.84	1.0	1.1
-0.30	-0.32	1.5	1.6
-0.83	-0.90	1.0	1.0

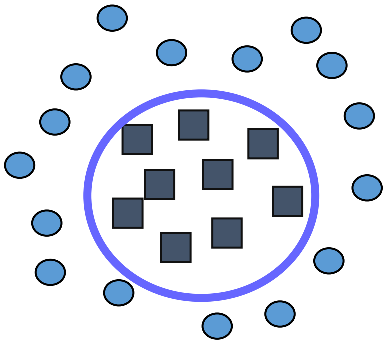
$$error = 0.245 = \frac{0.49}{2} = \frac{\lambda}{2}$$

DataRecovered = (*FeatureVector*(p=1) x *TransformedData*) + *OriginalMean*

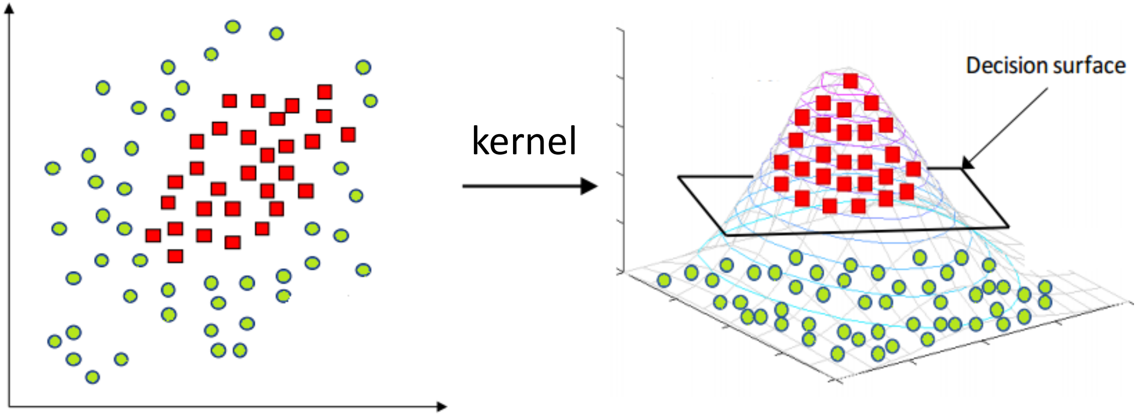
Outline

- Dimensionality reduction: why and how
- Feature selection
- **Linear transformations**
  - algebra ground
  - principal component analysis
  - compression and reconstruction
  - **non-linear kernels**
  - **linear discriminant analysis**
- Subspace selection
- Evaluation
- Data reduction

Nonlinearities using kernels



Linear projections will not detect this pattern



Nonlinear PCA using kernels

- Traditional PCA applies linear transformation (ineffective for nonlinear data)
- **Solution**: apply nonlinear transformation to potentially higher-dimensional spaces

$$\varphi: \mathbf{x} \rightarrow \varphi(\mathbf{x})$$

  - how? apply the kernel trick: PCA rewritten in terms of dot product

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \bullet \varphi(\mathbf{x}_j)$$

- **Example** $\varphi: (a_{i1}, a_{i2}, a_{i3}) \rightarrow (a_{i1}, \sqrt{a_{i2}a_{i3}}, a_{i1}{}^2 a_{i3})$

simplified: transform $A = \{\mathbf{x}_1, .., \mathbf{x}_n\}$ accordingly and apply PCA

$$\mathbf{x}_i = (2,4,1) \rightarrow (2,2,4)$$

Dimensionality reduction techniques

Linear transformations

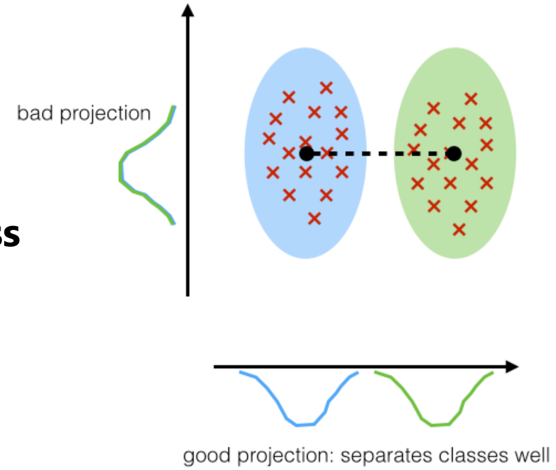
- **Eigenvalue analysis**: SVD, PCA and KL: O(nm^2) time
- **FastMap**: approximate searches in O(nm) time
  (optimal searches also in O(nm^2) time)
- **Multi-dimensional scaling**: O(nm^2) time
- **Random projections**: O(nm) time

Non-linear transformations

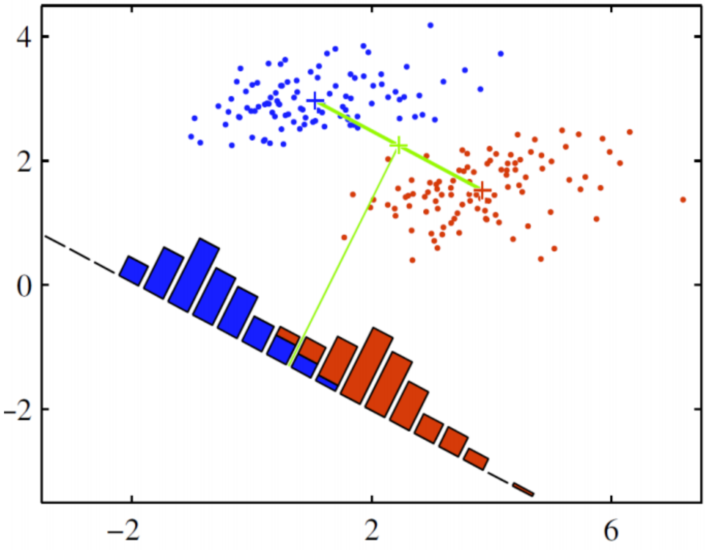
- **Kernel SVD/PCA**
- Locally linear embedding (**LLE**)
- Laplacian eigenmaps (**LEM**)
- Semidefinite embedding (**SDE**)

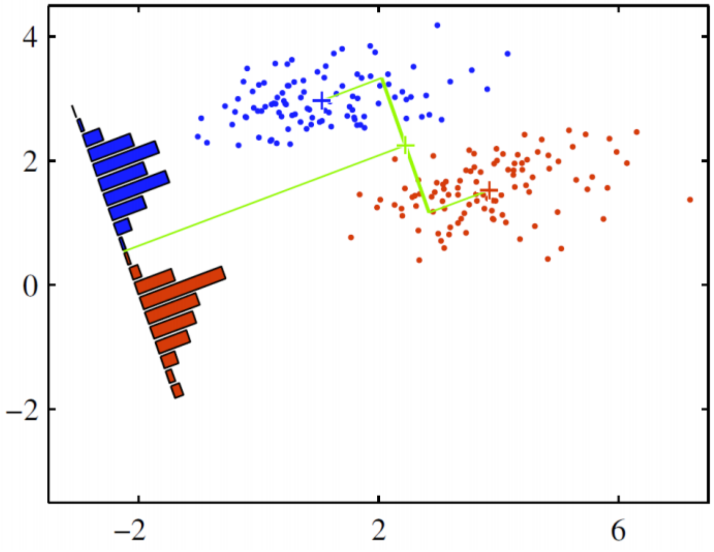
Linear discriminant analysis (LDA)

- Challenges of PCA in supervised settings?
  - Reduction does not consider impact on the ability to discriminate output variables
- Goal: data transformation guaranteeing class separation
- Principle: pick a new dimension that
  - **maximize separation between projected classes**
  - **minimize variance of observations within each class**
- Solution: **LDA**
  - eigenvectors based on between-class and within-class covariance matrices



Linear discriminant analysis (LDA)



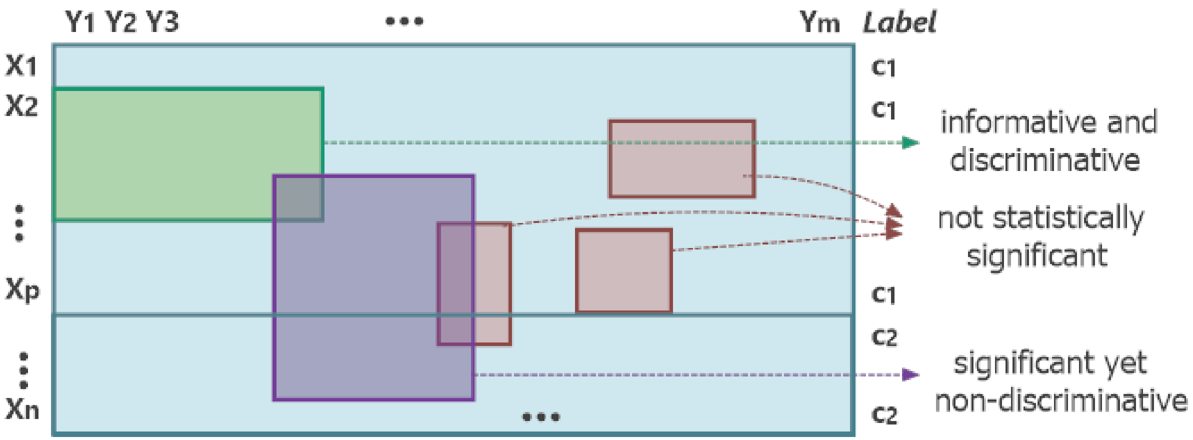


Outline

- Dimensionality reduction: why and how
- Feature selection
- Linear transformations
    - algebra ground
    - principal component analysis
    - compression and reconstruction
    - non-linear kernels
    - linear discriminant analysis
- **Subspace selection**
- Evaluation
- Data reduction

Subspace selection

- *Minimize overfitting*: remove uninformative regions (focus on informative/discriminative patterns only)
- *Minimize underfitting*: mine *all* relevant regions



(assumes reordering of rows and columns)

Subspace selection

- Addresses limitations of feature selection/extraction
  - single space → multiple compact spaces
  - goodness criteria computed from all observations → goodness verified on a subset of observations
- Yet limited applicability
  - **pattern mining** and **biclustering**
  - **associative classification**
  - pattern-centric visualization

Outline

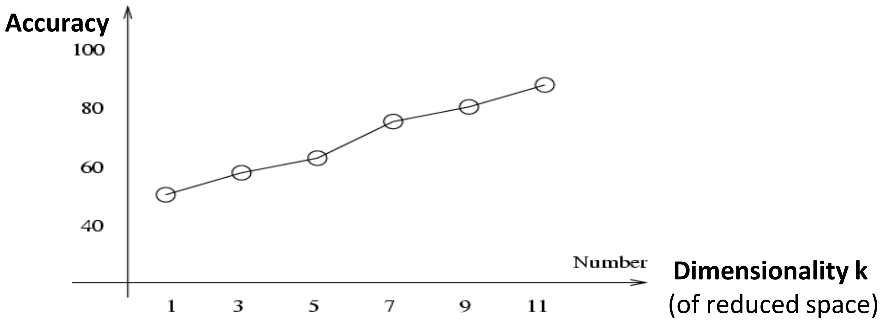
- Dimensionality reduction: why and how
- Feature selection
- Linear transformations
  - algebra ground
  - principal component analysis
  - compression and reconstruction
  - non-linear kernels
  - linear discriminant analysis
- Subspace selection
- **Evaluation**
- Data reduction

Evaluation of low-dimensional spaces

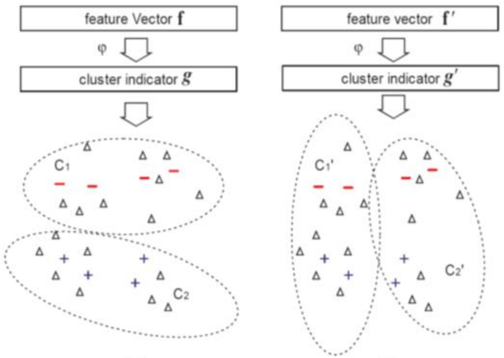
- Direct metrics
  - **match scores** against a reference set features/subspaces (only suitable for artificial data or data with prior knowledge)
- Indirect evaluation
  - **Learning on the original versus reduced data space** (e.g., predictive accuracy, goodness of resulting clusters)
    - before-and-after comparison
    - comparison using different learning algorithms
  - **Reconstruction error**
  - **Learning curves**

Evaluation of low-dimensional spaces

- learning curves



- before-and-after fitness using clustering



(a)
The cluster structure corresponding to culster indicator g

(b)
The cluster structure corresponding to culster indicator g'

TÉCNICO+
FORMAÇÃO AVANÇADA

Outline

- Dimensionality reduction: why and how
- Feature selection
- Linear transformations
  - algebra ground
  - principal component analysis
  - compression and reconstruction
  - non-linear kernels
  - linear discriminant analysis
- Subspace selection
- Evaluation
- **Data reduction**

Data reduction

Dimensionality reduction is one of the multiple forms of data reduction

- *data reduction* similarly considered to aid learning and time-memory efficiency

Other **data reduction strategies**:

- **domain reduction**: reduce data representation
  - *binning/discretization* of real-valued attributes, e.g. continuous variable onto ranges
  - *cardinality reduction* of nominal and ordinal attributes: aggregating categories
- **data size reduction (subsampling)**: reduce the number of instances
  1. *simple random subsample (SRS)*: randomly remove observations (either without or with replacement)
  2. *balanced sample*: remove observations guaranteeing that the reduced data satisfies a predefined criterion (e.g. balanced number of classes)

Data reduction

Subsampling (*continue*)

3. *stratified sample*: observations are divided into mutually disjointed parts, called strata, and removal (SRS) is done at each stratum
4. *data clustering*
   - clustering the data and use centroids instead of the actual data
5. *data condensation*
   - obtain minimal data set for correctly classifying all original observations
   - emerges from the fact that naive sampling (random or stratified sampling) is not suitable to learn from noisy data (dependent on algorithms)
6. *data squashing*
   - compress ("squash") data in such a way that a statistical analysis carried out on the compressed or original data yields the same outcome

Thank you!

Rui Henriques

rmch@tecnico.ulisboa.pt