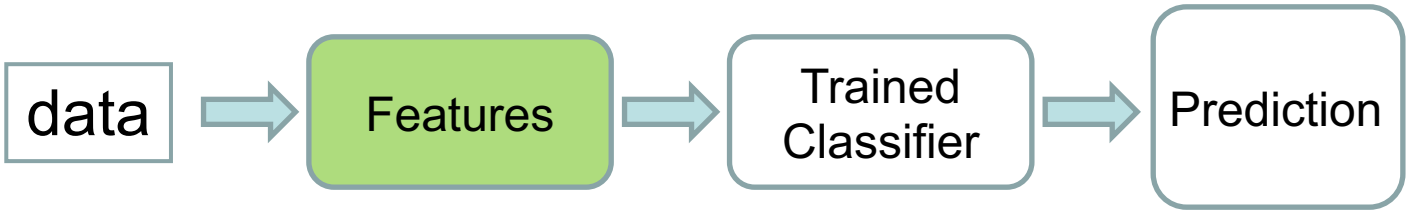
COMS20011 – Data-Driven Computer Science

Features

April 2023

Majid Mirmehdi

Next in DDCS



Feature Selection and Extraction

- Signal basics and Fourier Series
- 1D and 2D Fourier Transform
- **Another look at features**
- Convolutions

Examples of Features

- Primitive features, e.g.:
  - weight, length, width, height, volume …
  - amplitude, frequency, phase, duration, roll-off, flux …
  - beats per minute, temperature, pressure,...
  - edges, corners, lines, curvature, …
  - mean RGB colour, colour histogram, …
- Semantic features, e.g.:
  - colour layout (red, cyan, magenta,…)
  - texture descriptors (coarse, fine, rough, smooth,…)
  - shape descriptors (rectangular, circular, elliptical,…)
  - kind of day (warm, cold, sunny, rainy, …)
- Statistical features, e.g.:
  - mean, median, variance, percentiles, moments, ...
- . . .

Example: Image Features

Matching features (while also exploiting how they are arranged in the scene) would be much more efficient than matching all pixels

Common features between images allows us to perform tasks such as scene matching, face recognition, 3D model generation, and much more!

Quick Review: Features

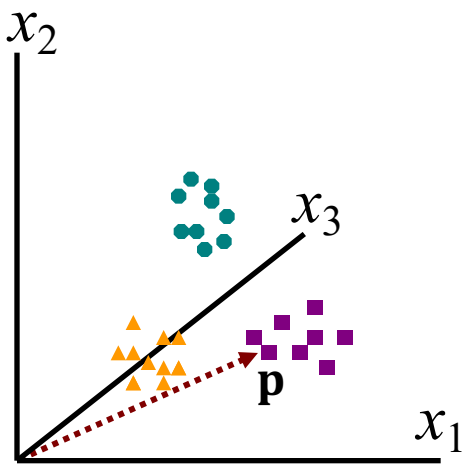
$$\mathbf{p} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_d \end{bmatrix}$$

- Features describe characteristics of our data.
- The combination of d features is represented as a d-dimensional column vector called a *feature vector*.
- The d-dimensional space defined by the feature vector is called the *feature space*.

$\mathbf{p}$ is a point in feature space X

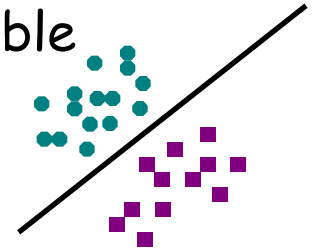
$$\mathbf{p} \in X$$

Example: 3D feature space X



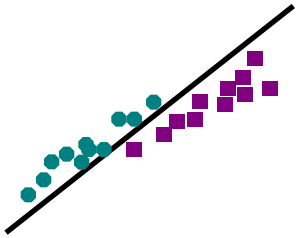
Feature Properties – *what makes a good feature vector?*

good, linearly-separable features

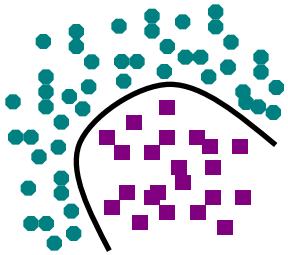


bad features

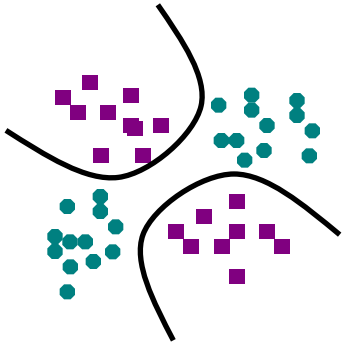
highly correlated features



nonlinearly-separable features



multimodal features



Dimensionality Reduction

- Strive for compact representation of the *properties* of data.
- This compact representation removes redundancy/irrelevancy.
- The choice of features is very important as it influences:
  - accuracy of classification
  - time needed for classification
  - no. of learning examples
  - difficulty in performing classification

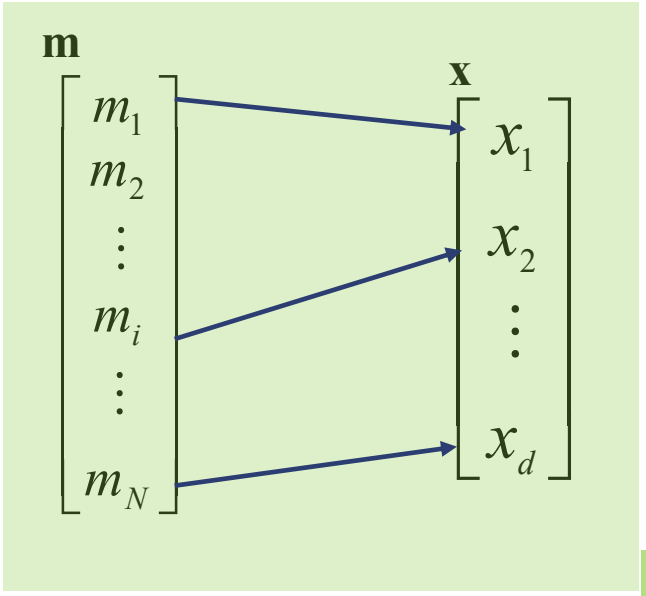
Feature Selection and Feature Extraction:

- to generate a set of characteristic attributes from data
- to allow representation of data in a *reduced dimension*

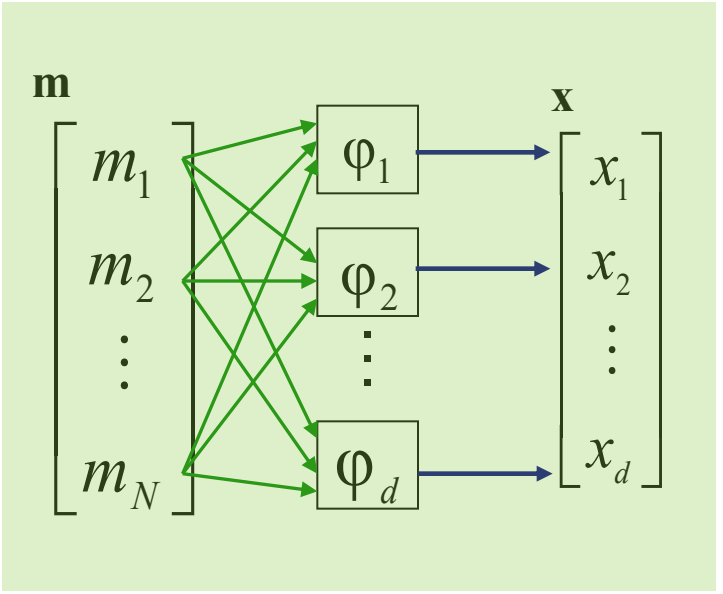
Selection or Extraction?

Two general approaches to dimensionality reduction:

- Feature Selection: Selecting a subset of the existing features without a transformation
- Feature Extraction: Transforming the existing features into a lower dimensional space



$d \ll N$



Implementing Feature Selection

- Feature Selection is necessary in a number of situations, e.g. there may be too many features or may be too expensive to obtain.
- Feature Selection involves a search strategy that may explore the space of all possible combinations of features.

Given a feature set $\{\boldsymbol{x}_i\}, i = 1, \ldots, N$, find a subset $\mathbf{X}$ of size d with $d < N$, that optimizes an objective function $J(\mathbf{X})$, e.g. P(correct classification).
This function would have to be evaluated many times:

e.g. for 10 features out of 25 one would still have to consider 3,268,760 feature sets.

$$\frac{N!}{(N-d)!d!}$$

Implementing Feature Selection

- Feature Selection is necessary in a number of situations, e.g. there may be too many features or may be too expensive to obtain.
- Feature Selection involves a search strategy that may explore the space of all possible combinations of features.

Given a feature set $\{\boldsymbol{x}_i\}, i = 1, \ldots, N$, find a subset $\mathbf{X}$ of size d with $d < N$, that optimizes an objective function $J(\mathbf{X})$, e.g. P(correct classification).

This function would have to be evaluated many times:

$$\frac{N!}{(N-d)!d!}$$

e.g. for 100 features out of 10000 one would have to consider K feature sets. What is K?

Implementing Feature Selection

K = 5,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,

And continue with as many more 0s as above for 322 more slides....

i.e. $K = 5 \times 10^{35101}$

Heuristic Feature Selection Methods

- Assume features are independent.
- Best single features can be chosen by significance tests.
- *bottom-up*: build up d features incrementally, **starting with an empty set** → *step-wise feature selection*:
  - The best single feature is picked first
  - Then next best feature conditioned to the first, ...
- *top-down*: **start with full set** of features and remove redundant ones successively → *step-wise feature elimination*

Feature Extraction

- Linear or non-linear transformation of the original variables to a lower dimensional feature space → also known as *feature selection in the transformed space*.
- Given a feature space R^N with feature vectors $\mathbf{m}$, find a mapping

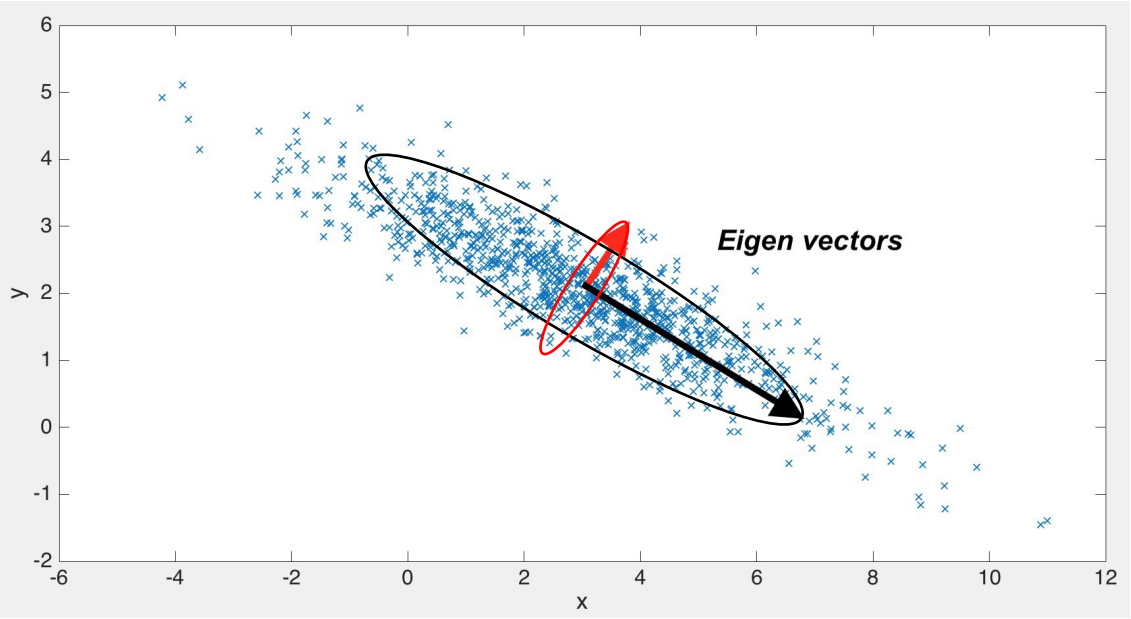
  $\mathbf{x} = \varphi(\mathbf{m}): R^N \Rightarrow R^d, d < N$, such that the transformed feature vector $\mathbf{x} = \{x_i\} \epsilon\, R^d$ preserves (most of) the information or structure in R^N.
- Principal Components Analysis (PCA) is an example of a transformed space for dimensionality reduction from which we can extract features (out of unit's scope)

Simple example: a transformed space where various representations of a class of objects would exhibit relevant features to identify all instances of that class of objects.

How would a robot recognise an object, given there might be translation, rotation, scaling, projective distortion, deformation, etc. of instances of the same class of object?

REMINDER – Covariance Matrix: Eigen analysis

- Eigenvectors and eigenvalues define **principal axes** and spread of points along directions
- **Major axis** - eigenvector corresponding to larger eigenvalue (i.e. larger variance)
- **Minor axis** - eigenvector corresponding to smaller eigenvalue (i.e. smaller variance)
- These can be represented using major and minor axes of ellipses



Dimensionality Reduction

- Key factor in good dimensionality reduction is to maintain as much of the variance as possible!

- Sum of the variances = sum of all eigenvalues = 100% of variance in original data

$$\sum_{i=1}^{N} \lambda_i$$

- The proportion of the variance that each eigenvector represents can be calculated by dividing the eigenvalue corresponding to that eigenvector by the sum of all eigenvalues.

Then the first d eigenvalues can be said to account for a fraction of the total variance in the data.

$$\frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{N} \lambda_i}$$

Example: how to account for a % of variance

- Around 2000 people were asked a set of questions about their *Internet use*. Let's say they asked each person 50 questions.

- There are therefore $N = \mathbf{50}$ **variables, making it a 50-dimensional dataset**. There will then be 50 eigenvectors and eigenvalues out of that dataset.

- Let's say the eigenvalues of the dataset were (in descending order):
  $39.8, 19.2, 17.0, 10.0, 3.2, 1.0, 0.4, 0.21, 0.0979, \ldots\ldots$ with a total sum of $\sum_{i=1}^{50} \lambda_i = 98.5$

- **Only 5 which have big enough values – indicating there is a lot of info (variance) along their corresponding five eigenvectors (directions)!**

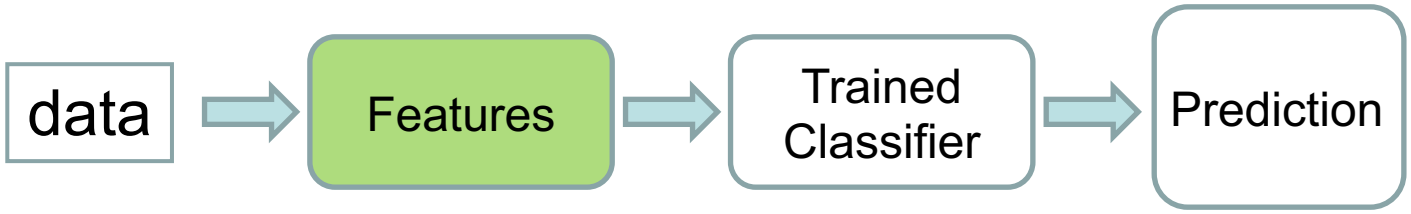
- The dataset can thus be reduced from 50 dimensions to only 5 by ignoring all the eigenvectors that have insignificant eigenvalues. Nice way of simplifying the data!

- **Percentage of variance captured by the first 5 components:**

$$\frac{\sum_{i=1}^{5} \lambda_i}{\sum_{i=1}^{50} \lambda_i} \Longrightarrow \frac{89.2}{98.5} \Rightarrow \sim 91\%$$

Adapted from: http://jurnal-singkat.blogspot.co.uk/2013/10/principal-component-analysis-4-dummies.html

Next in DDCS



Feature Selection and Extraction

- Signal basics and Fourier Series
- 1D and 2D Fourier Transform
- Another look at features
- **Convolutions**