MultiDimensional Signal Processing
Master Degree in “Ingegneria delle Telecomunicazioni”
A.A. 2017-2018

Pietro Guccione, PhD

DEI - DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL’INFORMAZIONE
POLITECNICO DI BARI

Pietro Guccione – Assistant Professor in Signal Processing
(piетro.guccione@poliba.it, http://dee.poliba.it/guccioneweb/index.html)
Lecture 3 - Summary

Pre-processing Steps

- Data cleaning
- Data integration and transformation
- Data reduction
- Sampling, discretization, normalization
Why Data Preprocessing?

Data in the real world are *dirty*
- incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
- noisy: containing errors or outliers
- inconsistent: containing discrepancies in codes or names

No quality data, no quality mining results
- Quality decisions must be based on quality data
- Data warehouse needs consistent integration of quality data
Multi-Dimensional Measure of Data Quality

• A well-accepted multidimensional view (from the most to the least important):
  o Accuracy (data must not be biased w.r.t. the physical quantity they claim to represent)
  o Completeness (data must be complete to interpret the physical quantity)
  o Consistency (increasing the number of data, a more precise representation of the real physical quantity or phenomenon must be given)
  o Timeliness (data must arrive/be processed in a usable time)
  o Believability (data must be reasonable)
  o Value added (data must represent significant information)
  o Interpretability (data must be clear to the user, in order to be processed)
  o Accessibility (data must be accessible)
Major Tasks in Data Preprocessing

• Data cleaning
  o Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies

• Data integration
  o Integration of multiple databases, data cubes, or files

• Data transformation
  o *Normalization* and aggregation

• Data reduction
  o Obtains *reduced representation* in volume but produces the same or similar analytical results

• Data discretization
  o Part of data reduction but with particular importance, especially for numerical data
Data Cleaning

Fill in missing values
Identify outliers and smooth out noisy data
Correct inconsistent data

Data mining operations often refers to **simple manipulations of data sheets**. **MV Signal Processing** instead makes data transformation and data reduction applying **mathematical operations** on data (domain transformation, linear combination, quantization, etc)
Data Integration

- Data integration:
  - combines data from *multiple sources* into a coherent store

- Schema integration
  - integrate metadata from different sources
  - Entity identification problem: identify real world entities from multiple data sources

- Detecting and resolving data *value conflicts*
  - for the same real world entity, attribute values from different sources may be different
  - possible reasons: different representations, different scales, (e.g., metric vs. British units)
Redundant data (which is a good things in PC networks…) occur often during integration of multiple databases:

- The same attribute may have different names in different database

Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Data coming from scientific experiment are not likely to occur in multiple copies, due to the presence of measurement noise (and because the experiment may be expensive and it is supposed completely under the control of the experimenter).

This can occur in experiment whose outcome is categorical or when the result is drawn. Anyway, for experiments taken in the same conditions and whose result is the same, a single copy can be kept.
Data Transformation

Main manipulation on data, to highlight properties of interest:

- **Smoothing**: remove noise from data
- **Normalization**: scaled to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- **Aggregation**: summarization, data cube construction
- **Generalization**: concept hierarchy climbing
Data Transformation: Smoothing

Smoothing consists in applying a local (moving) average on data. It is basically a low-pass filtering.

Basic filters that can be applied:
- Rectangular filtering (i.e. it makes the average of the last n samples)
- Triangular filtering (i.e. it makes the average of the last n samples, but reducing in weight the importance as the samples are far in the past)
- Exponential filtering, which introduces all the techniques of moving average

The moving average is a calculation to analyze data that creates a series of averages of the full data set. It is basically the output of a finite impulse response filter applied to the data when intended as a time series.

Variations include: simple, cumulative and weighted forms

**Simple**

The size N depends on the memory of the system or of the degree of smoothing we desire.

\[
SMA_n = \frac{x_n + x_{n-1} + \ldots + x_{n-N+1}}{N}
\]

\[
SMA_{n+1} = SMA_n + \frac{x_{n+1}}{N} - \frac{x_{n-N+1}}{N}
\]
Data Transformation: Smoothing

Cumulative
In the cumulative moving average we suppose that the previous average is somewhat related with the current one. So, we use the previous cumulated value:

\[ CMA_n = \frac{x_n + x_{n-1} + \ldots + x_{n-N+1}}{N} \]

\[ CMA_{n+1} = \frac{x_{n+1} + n \cdot CMA_n}{n + 1} \]

Exponential
In the exponential moving average the cumulation occurs through an infinite impulse response filter, since the weighting factors decrease exponentially (i.e. they never completely dampen)

\[ S_1 = X_1 \]

\[ S_n = \alpha \cdot X_n + (1 - \alpha) \cdot S_{n-1} \quad n > 1 \]

The coefficient \( \alpha \) (in \([0,1]\)) represents the degree of weighting decrease, a constant smoothing factor.
Data Transformation: Smoothing

Exponential

Smothes the data
Eliminates cyclic, seasonal and irregular movements
Loses the data at the beginning or end of a series
Sensitive to outliers (can be reduced by weighted moving average)

\[
S_1 = X_1
\]
\[
S_n = \alpha \cdot X_n + (1 - \alpha) \cdot S_{n-1} \quad n > 1
\]

...  
\[
S_n = \alpha \cdot \left[ X_n + (1 - \alpha) \cdot X_{n-1} + (1 - \alpha)^2 \cdot X_{n-2} + (1 - \alpha)^3 \cdot X_{n-3} + \ldots \right]
\]
Data Transformation: Smoothing

Evolution of Exponential Moving Average: The **Holt-Winters model**

The Holt-Winters model local linear trend may be extended to allow for seasonality forecasting procedure. It uses three smoothing constants (to be estimated), in the interval [0,1].

\[
S_n = \lambda_0 (X_n - c_{n-s}) + (1 - \lambda_0) (S_{n-1} + b_{n-1}) \\
b_n = \lambda_1 (S_n - S_{n-1}) + (1 - \lambda_1) b_{n-1} \\
c_n = \lambda_s (X_n - S_n) + (1 - \lambda_s) c_{n-s}
\]

Seasonal effect, \( s \) is the season period
Normalization is used to report data within a more reasonable interval. The operation is usually performed one variate at time.

Reasons:
- To have a fair comparison with the other variates
- To highlight the data variability (remove the mean)
- To report the data in an interval where a specific processing can be executed.

Example: data must be all positive

\[ X' = X - X \]  
Remove the mean

\[ X' = \frac{X - \text{min}(X)}{\text{max}(X) - \text{min}(X)}(b - a) + a \]  
Min-max normalization

\[ X' = \frac{X - \bar{X}}{\sigma_x} \]  
Z-scoring

\[ \begin{align*}
X' & = \frac{X - \text{perc}_{10}(X)}{\text{perc}_{90}(X) - \text{perc}_{10}(X)} \\
X'(X < \text{perc}_{10}(X)) & = \text{perc}_{10}(X) \\
X'(X > \text{perc}_{90}(X)) & = \text{perc}_{90}(X)
\end{align*} \]  
10-90 percentile normalization
Data Reduction Strategies

- Warehouse may store terabytes of data: Complex data analysis/mining may take a very long time to run on the complete data set (example: social data as tweets or physical data as astronomy DB)

- Data reduction:
  - Obtains a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results (it depends on the kind of analysis).

- Data reduction strategies:
  - Data cube aggregation
  - Dimensionality reduction
  - Numerosity reduction
  - Discretization and concept hierarchy generation.
Aggregates are used in multivariate data to produce effects on the time it takes to query large sets.

Basically, an aggregate is a summary table that can be derived by performing a grouping, based on simple criteria (in DB language: making a proper query).

A possible example is to take a dimension and change the granularity of this dimension. When changing the granularity of the dimension (values are quantized using less quantum values), the table is partially summarized to fit the new delta.

Another way is to remove a dimension (i.e. a variable) or by joining dimensions (using a common descriptor. Example: a single variate to describe both the geographical coordinates, after the reduction of granularity of latitude/longitude grid to a 1°x1° and the numbering of all the possible coordinates as sequential numbers).
The basic stages involved in the design of a classification system

Patterns (images, waveforms...)

sensor

feature generation & selection (dimensionality reduction)

classifier design

system evaluation

Reasons for Dimensionality Reduction: Design of a Classifier
Assume that we have access to an image database with a number of patterns, some of which are known to be originated from class A and others from class B.

What are the measurable (possibly, the minimum measurable) quantities that make these two regions distinct from each other?
Reasons for Dimensionality Reduction: Design of a Classifier

Example:

Generated features

benign lesion

malignant lesion
Reasons for Dimensionality Reduction: Design of a Classifier

Example:

Data points (images in the feature space)
Reasons for Dimensionality Reduction: Design of a Classifier

Example:

benign lesion

malignant lesion

Decision subspace
**Reasons for Dimensionality Reduction: Design of a Classifier**

Why dimensionality reduction?

- **Patterns** (images, waveforms…)
- **sensor**
- **feature generation & selection**
  - (dimensionality reduction)
- **classifier design**
- **system evaluation**

**Goal**: Find an alternative representation of the data (sounds, images …) to reduce the dimensionality and **retain the information that are relevant** for the classification task.

- **How are the features generated?** In the preceding example, we used the mean and the standard deviation, because we knew how the images had been generated. In practice, this is far from obvious and it is problem-dependent.
- **What is the best number of features to use?** A larger than necessary number of feature candidates is generated, and then the “best” of them are adopted.
Two approaches are available to perform dimensionality reduction:

- **Feature selection**: choosing a subset of all the features (the ones more informative)
- **Feature extraction**: creating a subset of new features by combination of the existing ones.

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N
\end{bmatrix}
\xrightarrow{\text{feature selection}}
\begin{bmatrix}
  x_{i1} \\
  x_{i2} \\
  \vdots \\
  x_{iM}
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N
\end{bmatrix}
\xrightarrow{\text{feature extraction}}
\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_M
\end{bmatrix} = f
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N
\end{bmatrix}
\]
Dimensionality Reduction

Let us consider a series of two-dimensional data:

For the current dataset:

\[ R = \begin{bmatrix} 1.0000 & -0.2074 \\ -0.2074 & 1.0000 \end{bmatrix} \]

The correlation between \( x_1 \) and \( x_2 \) is \(~20\%\), i.e. \(~20\%\) of the information in \( x_1 \) is also contained in \( x_2 \).

For standardized data:

\[ z = \frac{x - \bar{x}}{\sigma} \]

There is redundancy, but is little
In general, the optimal mapping $y = f(x)$ will be a non-linear function.

However, feature extraction is commonly limited to linear transforms: $y = Wx$

Some unsupervised techniques

Principal Component Analysis, Canonical Correlation Analysis, Mixtures of Probabilistic PCA, Kernel Methods…

In general, these methods find general types of relations in datasets. For many algorithms that solve these tasks, the data are transformed into feature vector representations via a proper map, or (as in kernel methods) a user-specified kernel over pairs of data points is used to make transformation.
Both heuristic feature selection methods:

- **Best single feature**: under the feature independence assumption (which is not the general case, unfortunately) chosen by significance tests (example: the one that produces the best classification. Information Gain: $\text{IG}(X,a) = H(X) - H(X|a)$)

- **Best step-wise feature selection**:
  - The best single-feature is picked first
  - Then, next best feature conditioned to the first, ...

- **Step-wise feature elimination**:
  - Repeatedly eliminate the worst feature

- **Best combined feature selection and elimination**: 
A data reduction technique. Data are represented for their counting in the interval of their dynamics.

It is used to estimate the Probability density function of a set of data, provided that:

- Data are extracted from a given random variable.
- All the data follow the same r.v. with the same parameters and during the data extraction, the parameters have not changed (stationarity of dataset).
- The set of samples is representative of the whole data dynamics (i.e. all the data are sufficient to represent the statistics: the dataset is also ergodic).

Problem in estimation:

- Determine the correct set of intervals to make a suitable representation.
- Determine the size of the intervals (equal/unequal).
Example: samples from a Gaussian r.v. with 0 mean and std = 1. N=100

Histogram representation using equal intervals, $K = 8$

Histogram representation using equal intervals, $K = 12$
Histogram

Example: samples from a Gaussian r.v. with 0 mean and std = 1. N=1000

Histogram representation using equal intervals, $K = 12$

Histogram representation using equal intervals, $K = 20$
Example: samples from a Gaussian r.v. with 0 mean and std = 1. N=10000

Histogram representation using equal intervals, $K = 128$

Preliminary knowledge of the dynamics is important to set the interval and to decide a proper representation of the samples in suitable intervals. Example: if we would have known that the samples are drawn from a Gaussian r.v. with zero-mean and unitary standard deviation, we would have decided for an optimal interval going from -4 to 4, if a large set of data is present, from -3 to 3 for reduced set (unlikely that a value over -4 or +4 is present. Compute the chance).
Example: samples from the square of a Gaussian r.v. with mean -4 and std = 0.5. N=10000

Histogram representation using equal intervals, $K = 128$

Lacking in the knowledge of a reasonable interval of representation

Saturation at the highest interval, of the remaining samples
Example: samples from the square of a Gaussian r.v. with $-4$ mean and std $= 0.5$. $N=10000$

Histogram representation using equal intervals, $K = 128$

\[ \text{pdf is not realistic, since the area under the curve must be 1.} \]

Normalization for the number of samples (hist makes the counting)

\[
\begin{align*}
\text{>> } [H,X] &= \text{hist}(y,(0:0.5:35)); \\
\text{>> } H &= H/N; \\
\text{>> figure, plot}(X,H)
\end{align*}
\]
Histogram

Can be used to make comparison between:
- Two sample pdf (i.e. two histograms achieved from two different datasets)
- A sample pdf and a theoretical pdf with a given set of parameters (the parameters may be estimated from samples or anyway given).

Example: visual comparison between the set of samples and a Gaussian curve with zero mean and unitary standard deviation.

```matlab
N=10000;  
x = randn(N,1);  
[H,X] = hist(x,(-3:0.1:3));  
H=H/sum(H);  
figure,bar((-3:0.1:3),H)  
hold on;  
y = 1/sqrt(2*pi)*exp(-0.5*(-3:0.1:3).^2);  
y=y/sum(y);  
plot((-3:0.1:3), y, 'r', 'LineWidth', 2);
```
Cluster analysis, which is the most well-known example of unsupervised Learning.

The methodology consists of various algorithms that try to organize a data set into homogeneous subgroups, with the rule that data within a subgroup are more “close” (distance is properly defined) that data in other groups.

The center of mass of the group can be used as a representative value for the cluster (natural quantization).
Clustering

Rarely clustering of data is clear as in the example below.

It is likely to have such situation.
After clusterization, we can identify a “representative member” or the center of mass of the data within a group (the centroid).

Successive data, if such representation is reliable, can be identified using a partition of the variable space (Voronoi regions, that correspond to distance to centroids).
Discretization is the way the data are divided into a finite (discrete) set to have a more compact representation.

The values the data take into each variate must be divided into proper intervals. For univariate data, discretization is basically a 1-D quantization. Main difference with clustering is that in the discretization we do not have to “discover” the number of intervals (clusters), nor their spacing. Usually, in the discretization, the intervals are equally distributed through the dynamics of the variate.

After discretization, interval labels can be used to replace the actual data values.
Discretization

It is used to reduce the data size and prepare for further analysis.

Difficulties:
- Estimate the dynamic of variation of data (accounting for possible presence of outliers),
- Decide the intervals within the dynamic,
- To deal with multivariate data (intervals become volume in n-D hyperspaces),
- To deal with combination of categorical-numerical data

Discretization can be done in a hierarchic way, dealing with more level of replacing and data reduction
In statistics, the **sampling** of a population is the problem of selecting a subset of samples, so that the samples are representative of the whole population.

Sampling is somewhat connected with the concept of clustering. If we would choose data from a single region only, those data would not be representative of the whole population.

Sampling concerns the problem of estimating (or supposing) properties of the population so to make a suitable selection with a reduced set.
The sampling process comprises several stages:

- Defining the population. This deals with the identification of the variates and of the dynamic of each variate, if possible.
- Specification of the sampling frame, if possible (example: the list of all the data of the whole population, when possible).
- Specify a sampling method to select items.
- Determination of the sample size.
- Make the sampling and make data collection.

**Stratified sampling:**
It is a way to sample each subpopulation independently, so to get samples that are proportional to the size of each subpopulation. Sub-groups must be known before sampling. Stratification is the process with which data are re-organized in subgroups. The different groups must have null intersection and must be exhaustive, i.e. they must form a partition of the whole set. Simple random sampling is then applied within each stratum.
Sampling

Raw Data

SRSWOR
(simple random sample without replacement)

SRSWR
Pre-processing Steps: Summary

- Multivariate data are incomplete, noisy and inconsistent. Data must be then: Cleaned (to remove noise or errors),
- Integrated (to add missing data),
- Outliers must be removed (to avoid future error in data interpretation),
- Normalized, to account for different dynamic of the same variate on more datasets and allows proper comparisons,
- Properly transformed, to account for different problems.

Transformation include:
- Normalization
- Smoothing
- Generation of statistics (such as histograms)
- Sampling, that allow a proper data selection within a population
- Clustering, to discover the representative sub-groups
- Discretization, to make data reduction using labels that replace data.