Chapter IV: Preprocessing

Knowledge Discovery in Databases

Luciano Melodia M.A.
Evolutionary Data Management, Friedrich-Alexander University Erlangen-Nürnberg
Summer semester 2021

Chapter IV: Preprocessing

This is our agenda for this lecture:

- **Data preprocessing: an overview.**
  - Data quality.
  - Major tasks in data preprocessing.
- Data cleaning.
- Data integration.
- Data reduction.
- Data transformation and data discretization.
- Summary.

Data quality: why preprocess the data?

This is our agenda for this lecture:

Measures for data quality: A multidimensional view:

- **Accuracy:** correct or wrong, accurate or not.
- **Completeness:** not recorded, unavailable.
- **Consistency:** some modified but some not, dangling refs, etc.
- **Timeliness:** timely updated?
- **Believability:** how trustworthy is it, that the data is correct?
- **Interpretability:** how easily can the data be understood?
- And even many more!

Major tasks in data preprocessing

Data cleaning:
- Fill in missing values.
- Smooth noisy data.
- Identify or remove outliers.
- Resolve inconsistencies.

Data integration:
- Integration of multiple databases.
- Data cubes or files.

Data reduction:
- Dimensionality reduction.
- Numerosity reduction.
- Data compression.

Data transformation and data discretization:
- Normalization.
- Concept-hierarchy generation.

Chapter IV: Preprocessing

- Data preprocessing: an overview.
  - Data quality.
  - Major tasks in data preprocessing.
- **Data cleaning.**
- Data integration.
- Data reduction.
- Data transformation and data discretization.
- Summary.

Data cleaning

Data in the real world is dirty. Lots of potentially incorrect data:

- E.g. instrument faukty, human or computer error, transmission error.
- **Incomplete:** lacking attributes, lacking certain attributes of interest or containing aggregate data.
  - E.g. occupation = "" (missing data).
- **Noisy:** containing noise, errors or outliers.
  - Stochastic deviation, imprecision.
  - E.g. measurements.
- **Inconsistencies:** containing discrepancies in codes or names.
  - E.g. age = "42", birthday = "03/07/2010".
  - Was rating "1,2,3" and now it is "A,B,C".
  - Discrepancy between duplicate records (e.g. address old and new).
- **Intentional** (only default value, e.g. disguised missing data):
  - Jan. 1 as everyone's birthday?

Incomplete (missing) data

Data is not always available.

- E.g. many tuples have no recorded value for several attributes.
- Examples are customer income in sales data.

Missing data may be due to:

- Equipment malfunction.
- Inconsistency with other recorded data and thus deleted.
- Data not entered due to misunderstanding.
- Certain data may not be considered important at the time of entry.
- Not registered history or changes of the data.

Missing data may need to be inferred.

How to handle missing data?

Ignore the tuple:

- Usually done when class label is missing (when doing classification).
- Not effective when the percentage of missing values per attribute varies considerably.

Fill in the missing value manually.

- Tedious or infeasible.

Fill in automatically with:

- A global constant, e.g. "unkown", maybe a new class.
- The attribute mean.
- The attribute mean for all samples belonging to the same class.
- **The most probable value:** Inference-based such as Bayesian formula or decision tree.

Noisy data?

Noise:

Random error or variance in a measured variable.
Stored value a little bit off the real value, up or down.
Leads to (slightly) incorrect attribute values.

May be due to:

Faulty or imprecise data-collection instruments.
Data-entry problems.
Data-transmission problems.
Technology limitation.
Inconsistency in naming conventions.

How to handle noisy data?

Beginning:

First sort data and partition into (equal-frequency) bins.
Then smooth by bin mean, by bin median or by bin boundaries.

Regression:

Smooth by fitting the data to regression functions.

Clustering:

Detect and remove outliers.

Combined computer and human inspection:

Detect suspicious values and check by human.
E.g. deal with possible outliers.

Data cleaning as a process

Data-discrepancy detection:

- Use **metadata** (e.g. domain, range, dependency, distribution).
- Check field overloading.
- Check uniqueness rule, consecutive rule and null rule.
- Use commercial tools:
  - **Data scrubbing:** use simple domain knowledge (e.g. postal code, spell-check) to detect errors and make corrections.
  - **Data auditing:** by analyzing data to discover rules and relationsships to detect violators (e.g. correlation and clustering to find outliers).

Data migration and integration:

- Data-migration tools: allow transformations to be specified.
- ETL (Extraction/Transformation/Loading) tools: allow users to specify transformations through a graphical user interface.

Integration of the two processes.

- Iterative and interactive (e.g. the Potter's Wheel tool).

Chapter IV: Preprocessing

- Data preprocessing: an overview.
  - Data quality.
  - Major tasks in data preprocessing.
- Data cleaning.
- **Data integration.**
- Data reduction.
- Data transformation and data discretization.
- Summary.

Data integration

Data integration:

Combine data from multiple sources into a coherend store.

Schema integration:

E.g. `A.cust-id` $\equiv$ `B.cust-#`.

Integrate metadata from different sources.

Entity-identification problem:

Identify the same real-world entities from multiple data sources.

E.g. Bill Clinton = William Clinton.

Detecting and resolving data-value conflicts:

For the same real world entity, attribute values from different sources are different.

Possible reasons:

- Different representations (coding).
- Different scales, e.g. metric vs. British units.

Handling redundancy in data integration

Redundant data often occur when integrating multiple databases.

- **Object (entity) identification:**
  The same attribute or object may have different names in different databases.
- **Derivable data:**
  One attribute may be a "derived" attribute in another table. E.g. annual revenue.

Redundant attributes:

- Can be detected by **correlation analysis** and **covariance analysis**.

Careful integration of the data from multiple sources:

- Helps to reduceavoid redundancies and inconsistencies and improve mining speed and quality.

Correlation analysis for nominal data (I)

Two attributes:

A has n distinct values: $A := \{a_1, a_2, \ldots, a_n\}$.
B has m distinct values: $B := \{b_1, b_2, \ldots, b_m\}$.

Contingency table:

Columns: the n values of A.
Rows: the m values of B.
Cells: counts of records with
$A' = \{a_i \in A \ : \ a_i = a_k \text{ for } a_k \in A\}$ and
$B' = \{b_j \in B \ : \ b_j = b_l \text{ for } b_l \in B\}$.

Expected count in cell (i, j):

$$e_{ij} = \frac{\#A' \cdot \#B'}{\#A + \#B}, \tag{1}$$

where $\#A + \#B$ is the total number of records.

Correlation analysis for nominal data (II)

χ^2-test:

$$\chi^2 = \sum_{i=1}^{N} \frac{(x_i - \hat{x}_i)^2}{\hat{x}_i}. \tag{2}$$

Summing over all cells of the contingency table.

No correlation (i.e. independence of attributes) yields χ^2 value of zero.

The larger the χ^2 value, the more likely the variables are related.

The cells that contribute the most to the χ^2 value are those whose actual count is very different from the expected count.

Correlation does not imply causality!

E.g. $\#$ of hospitals and $\#$ of car-thefts in a city are correlated.

Both are causally linked to the third variable: population.

ξ^2 calculation: an example

	Play chess	Not play chess	Sum (row)
Like Science fiction	250(90)	200(360)	450
Not like science fiction	50(210	1000(840)	1050
Sum (column)	300	1200	1500

Numbers in parenthesis are expected counts calculated based on the data distribution in the two categories.

χ^2 calculation:

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93. \tag{3}$$

It shows that "like science fiction" and "play chess" are correlated in the group.

Correlation analysis of numerical data

Correlation coefficient:

Also called Pearson's product-moment coefficient

$$r_{A,B} = \frac{\sum_{i=1}^{N+M}(a_i - \mu_A)(b_i - \mu_B)}{(N+M-1)\sigma_A\sigma_B} = \frac{\sum_{i=1}^{N+M}(a_i b_i) - (N+M)\mu_A\mu_B}{(N+M-1)\sigma_A\sigma_B}. \tag{4}$$

where $N = \#A$, $M = \#B$, μ_A and μ_B are the means of A and B, respectively. σ_A and σ_B denote the corresponding standard deviations.

If $r_{A,B} > 0$, A and B are positively correlated (A's values increase with B's).
The higher, the stronger the correlation.

$r_{A,B} = 0$: independent.

$r_{A,B} < 0$: negatively correlated.

Visually evaluating correlation

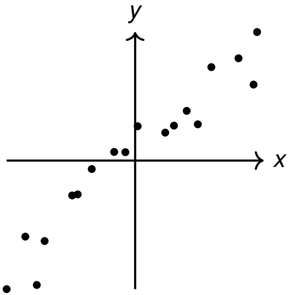


Figure: a) Positive correlation.

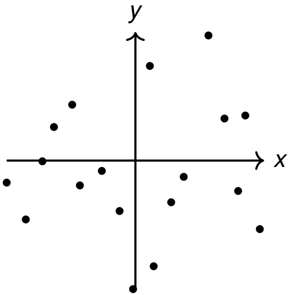


Figure: b) Uncorrelated/no correlation.

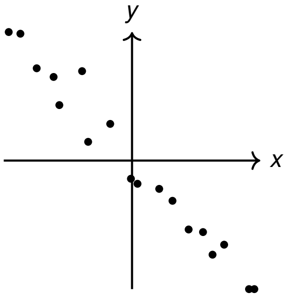


Figure: c) Negative correlation.

Covariance of numerical data (I)

Covariance is similar to correlation:

$$\mathrm{Cov}(A, B) = \frac{\sum_{i=1}^{n}(a_i - \overline{A})(b_i - \overline{B})}{n} \tag{5}$$

Pearson's correlation coefficient:

$$r = \frac{n\sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} a_i \sum_{i=1}^{n} b_i}{\sqrt{\left(n\left(\sum_{i=1}^{n} a^2\right) - \left(\sum_{i=1}^{n} a_i\right)^2\right)\left(n\left(\sum_{i=1}^{n} b^2\right) - \left(\sum_{i=1}^{n} b_i\right)^2\right)}}, \tag{6}$$

where n is the number of tuples.

Covariance of numerical data (II)

Positive covariance:

If $\mathrm{Cov}(A, B) > 0$, then A and B tend to be either both larger or both smaller than their expected values.

Negative covariance:

If $\mathrm{Cov}(A, B) < 0$, then if A is larger than its expected value, B is likely to be smaller than its expected value and vice versa.

Independence:

$\mathrm{Cov}(A, B) = 0$.

But the converse is not true: Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence.

Covariance: an example (I)

Can be simplified in computation as:

$$\begin{aligned} \mathrm{Cov}(A, B) &= E(A - E(A))(B - E(B)) && (7) \\ &= E(AB - AE(B) - E(A)B + E(A)E(B)) && (8) \\ &= E(AB) - E(A)E(B) - E(A)E(B) + E(A)E(B) && (9) \\ &= E(AB) - E(A)E(B). && (10) \end{aligned}$$

Covariance: an example (II)

Suppose two stocks A and B have the following values within some time: $(2,5), (3,8), (5,10), (4,11), (6,14)$.

If the stocks are affected by the same industry trends, will their prices rise or fall together?

$$E(A) = \frac{2+3+5+4+6}{5} = \frac{20}{5} = 4. \tag{11}$$

$$E(B) = \frac{5+8+10+11+14}{5} = \frac{48}{5} = 9.6. \tag{12}$$

$$\mathrm{Cov}(A,B) = \frac{2\cdot 5 + 3\cdot 8 + 5\cdot 10 + 4\cdot 11 + 6\cdot 14}{5} - 4\cdot 9.6 = 4. \tag{13}$$

Thus, A and B rise together since $\mathrm{Cov}(A,B) > 0$.

Chapter IV: Preprocessing

- Data preprocessing: an overview.
  - Data quality.
  - Major tasks in data preprocessing.
- Data cleaning.
- Data integration.
- **Data reduction.**
- Data transformation and data discretization.
- Summary.

Data reduction I: dimensionality reduction

Curse of dimensionality:

- When dimensionality increases data becomes increasingly sparse.
- Density and distance between points, which are critical to clustering and outlier analysis become less meaningful.
- The possible combinations of subspaces will grow exponentially.

Dimensionality reduction:

- Avoid the curse of dimensionality.
- Help eliminate irrelevant features and reduce noise.
- Reduce time and space required in data mining.
- Allow easier visualization.

Dimensionality-reduction techniques:

- Wavelet transforms.
- Principal component analysis.
- Supervised and nonlinear techniques (e.g. feature selection).

Data reduction strategies

Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) results.

Why data reduction?

- A database/data warehouse may store terabytes of data.
- Complex data analysis may take a very long time to run on the complete data set.

Data reduction strategies:

- Dimensionality reduction, i.e. remove unimportant attributes.
  - Wavelet transforms.
  - Principal component analysis.
  - Attribute subset selection or attribute creation.
- Numerosity reduction:
  - Regression and log-linear models.
  - Histograms, clustering and sampling.
  - Data cube aggregation.
- Data compression.

Mapping data to a new space

Fourier transform.

Wavelet transform.

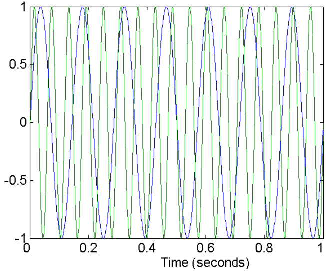


Figure: Two sine waves.

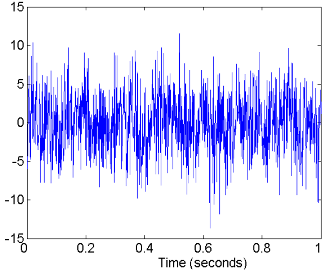


Figure: Two sine waves with noise.

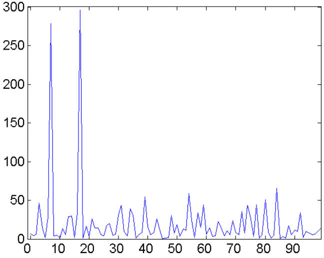


Figure: Frequencies.

What is wavelet transform?

Decomposes a signal into different frequency subbands.

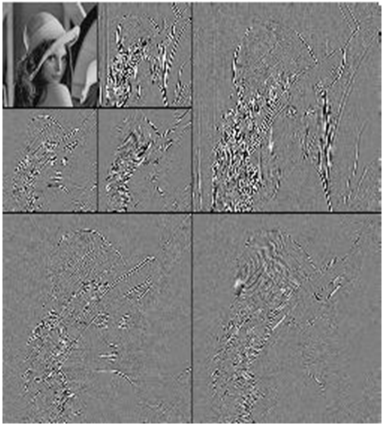
Applicable to *n*-dimensional signals.

Data transformed to preserve relative distance between objects at different levels of resolution.

Data transformed to preserve relative distance between objects at different levels of resolution.

Allow natural clusters to become more distinguishable.

Used for image compression.



Wavelet transformation

Discrete wavelet transform:
Transforms a vector X into a different vector X' of wavelet coefficients with the same length.

Compressed approximation:
Store only a small fraction of the strongest of the wavelet coefficients.

Similar to discrete fourier transform, but better lossy compression, localized in space.

Method:

- The length of the vector must be an integer power of 2 (padding with 0's if necessary).
- Each transform has two functions: smoothing and difference.
- Applied to pairs of data, resulting in two sets of data with half the length.
- The two functions are applied recursively until reaching the desired length.

Wavelet decomposition

Example:

$$X = \big(2, 2, 0, 2, 3, 5, 4, 4\big), \text{ can be transformed to} \tag{14}$$

$$X' = \big(2.75, -1.25, 0.5, 0, 0, -1, -1, 0\big). \tag{15}$$

Compression:
Many small detail coefficients can be replaced by 0's,
and only the significant coefficients are retained.

Resolution	Averages	Detail coefficients
8	$\big(2, 2, 0, 2, 3, 5, 4, 4\big)$	-
4	$\big(2, 1, 4, 4\big)$	$\big(0, -1, -1, 0\big)$
2	$\big(1\frac{1}{2}, 4\big)$	$\big(\frac{1}{2}, 0\big)$
1	$\big(2\frac{3}{4}\big)$	$\big(1\frac{1}{4}\big)$

Why wavelet transform?

Use hat-shaped filters:

Emphasize region where points cluster.
Suppress weaker information in their boundaries.

Effective removal of outliers:

Insensitive to noise, insensitive to input order.

Multi-resolution:test

Detect arbitrary shaped clusters at different scales.

Efficient: Complexity $\mathcal{O}(N)$.

Principal component analysis (I)

Principal component analysis is a method of summarizing the properties of a set of multivariate data samples.

It is a **linear transformation** method that is often used for data analysis or data compression.

Principal component analysis is often also called **Karhunen-Loeve transformation**.

PCA is equivalent to maximization of the information at the output of a neural network with linear neurons.

The goal of the principal component analysis (PCA) is the identification of n **normed orthogonal vectors** $\{x_i \in \mathbb{R}^m \mid i = 1, 2, \ldots, n\}$ within the input space, which **represent most of the variance** of the data.

Principal component analysis (2)

Consider sample vectors $u_1, u_2, \ldots, u_n \in \mathbb{R}^m$ centered at zero and a complete orthonormal system $\{x_i \in \mathbb{R}^m \mid i = 1, 2, \ldots n\}$, such that:

$$\langle x \rangle = \int x p(x) d^m x = 0, \tag{16}$$

$$||u|| = 1, \tag{17}$$

$$\langle x^T u \rangle = 0, \tag{18}$$

where the Euclidean norm of the vector is given by

$$||u_i|| = \left(\sum_{i=1}^{m} u_i^2 \right)^{\frac{1}{2}}. \tag{19}$$

Goal: find u^* such that $\langle (x^T u^*)^2 \rangle$, the variance of the projections of x onto u^* becomes maximal according to the probability distribution $p(x)$.

Principal component analysis (III)

Principal component analysis (IV)

Principal component analysis (V)

Principal component analysis (VI)

Principal component analysis (VII)

References (I)

D. P. Ballou and G. K. Tayi: Enhancing data quality in data warehouse environments. Comm. of ACM, 42:73-78, 1999.

A. Bruce, D. Donoho, and H.-Y. Gao: Wavelet analysis. IEEE Spectrum, Oct. 1996.

T. Dasu and T. Johnson: Exploratory Data Mining and Data Cleaning. John Wiley, 2003.

J. Devore and R. Peck: Statistics: The Exploration and Analysis of Data. Duxbury Press, 1997.

H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita: Declarative data cleaning: Language, model, and algorithms. VLDB'01.

M. Hua and J. Pei: Cleaning disguised missing data: A heuristic approach. KDD'07.

H. V. Jagadish et al.: Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), Dec. 1997.

References (2)

H. Liu and H. Motoda (eds.): Feature Extraction, Construction, and Selection: A Data Mining Perspective. Kluwer Academic, 1998.

J. E. Olson. Data Quality: The Accuracy Dimension. Morgan Kaufmann, 2003.

D. Pyle: Data Preparation for Data Mining. Morgan Kaufmann, 1999.

V. Raman and J. Hellerstein: Potter's Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB'01.

T. Redman: Data Quality: The Field Guide. Digital Press (Elsevier), 2001.

R. Wang, V. Storey, and C. Firth: A framework for analysis of data quality research. IEEE Trans. Knowledge and Data Engineering, 7:623-640, 1995.

Thank you for your attention.
Any questions about the forth chapter?

Ask them now, or again, drop me a line:
`luciano.melodia@fau.de`.