Queen Mary
University of London

ECS766 Data Mining

Week 3: Data Preprocessing

Emmanouil Benetos
emmanouil.benetos@qmul.ac.uk

October 2021

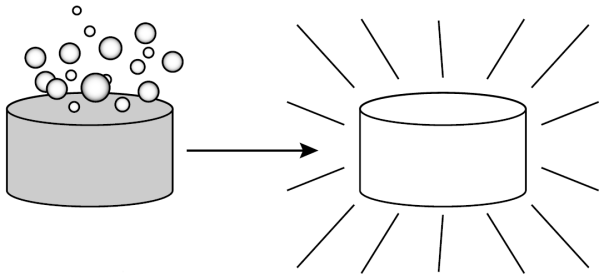
School of EECS, Queen Mary University of London

Last week: Data

- Attributes and Objects
- Characteristics of Data
- Types of Data
- Data Quality
- Basic Statistical Descriptions of Data
- Similarity and Distance

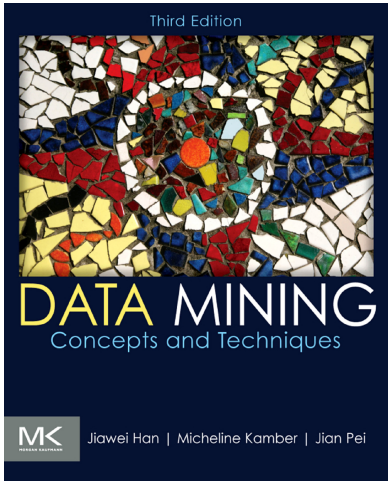
This week's contents

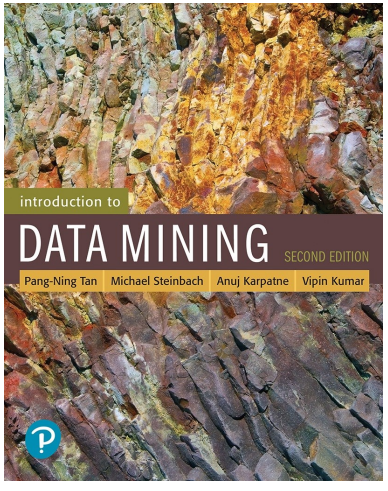
1. Data Preprocessing: An Overview
2. Data Cleaning
3. Data Integration
4. Data Reduction
5. Data Transformation and Data Discretisation

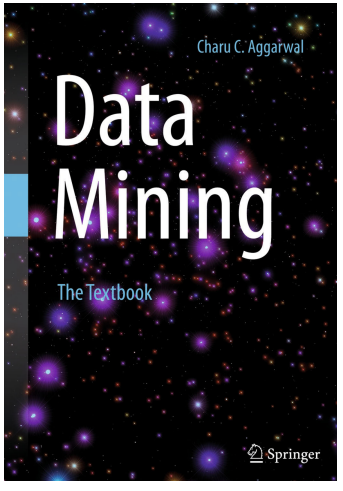


Reading

- **Chapter 3 of J. Han, M. Kamber, J. Pei, "Data Mining: Concepts and Techniques", 3rd edition, Elsevier/Morgan Kaufmann, 2012**
- Section 2.3 of P.-N. Tan, M. Steinbach, A. Karpatne, V. Kumar, "Introduction to Data Mining", 2nd edition, Pearson, 2019
- Chapter 2 of C. C. Aggarwal, "Data Mining: The Textbook", Springer, 2015







Data Preprocessing: An Overview

What is Data Preprocessing? — Major Tasks

- **Data cleaning**: handle missing data, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**: Integration of multiple databases, data cubes, or files
- **Data reduction**:
  - Dimensionality reduction
  - Numerosity reduction
- **Data transformation**:
  - Normalisation
  - Data discretisation
  - Concept hierarchy generation

Why Preprocess the Data?

Data have quality if they satisfy the requirements of the intended use. There are many factors comprising data quality:

- Accuracy: incorrect attribute values
- Completeness: not recorded, unavailable
- Consistency: data not consistent with other recorded data
- Timeliness: data not updated in a timely fashion
- Believability: how much the data is trusted by users
- Interpretability: how easily the data can be understood

Data Cleaning

Data Cleaning

Data in the real world is incomplete, noisy, and inconsistent.

- Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
  e.g. *Occupation = " "* (missing data)
- Noisy: containing noise, errors, or outliers
  e.g. *Salary = "-10"* (an error)
- Inconsistent: containing discrepancies in records
  e.g. *Age = "42"*, *Birthday = "03/07/2010"*
  e.g. Rating was *"1, 2, 3"*, now rating is *"A, B, C"*
- Intentional (disguised missing data)
  e.g. *Jan. 1 as everyone's birthday*

Missing Data

Data is not always available, and many entries might have no recorded value for several attributes.

Missing data may be due to:

- Equipment malfunction
- Being inconsistent with other recorded data and thus deleted
- Certain data may not be considered important at the time of entry
- Did not register 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 – not effective when the % of missing values per attribute varies considerably
- **Fill in the missing value manually**: tedious + infeasible?
- **Fill in the missing value automatically** with:
  - A global constant, e.g. “unknown” or “$-\infty$”– a new class?!
  - The attribute mean/median
  - The attribute mean/median for all samples belonging to the same class
  - The most probable value: inference-based using regression methods

In some cases, a missing value may not imply an error in the data!

Noisy Data

Noise: random error or variance in a measured variable

Incorrect attribute values may be due to:

- Faulty data collection instruments
- Data entry problems
- Data transmission problems
- Technology limitation
- Inconsistency in naming convention

How to Handle Noisy Data?

Binning

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

Sorted data for *price* (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into (equal-frequency) bins:

Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

Smoothing by bin means:

Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

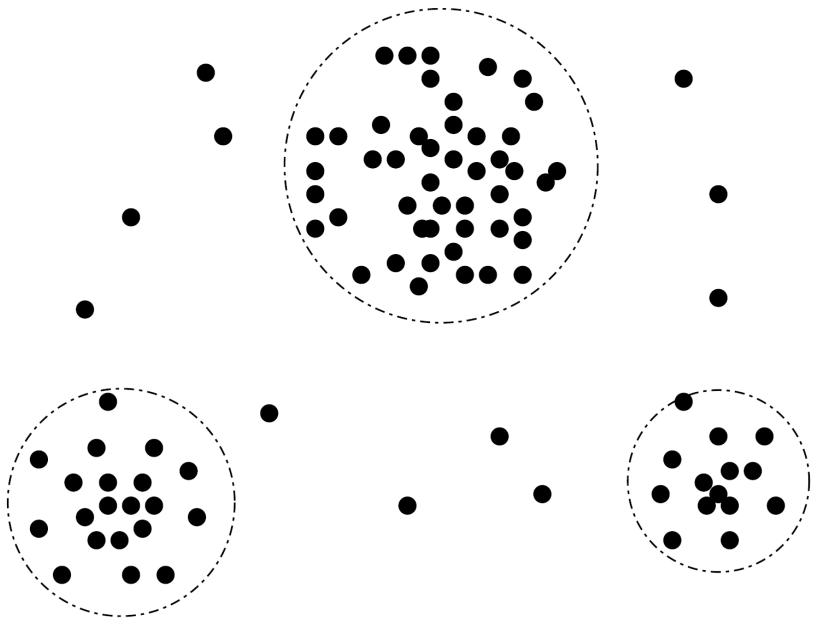
Smoothing by bin boundaries:

Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

How to Handle Noisy Data?

Regression: smooth by fitting the data into regression functions.

Outlier Analysis: detect and remove outliers.



Semi-supervised: combined computer and human inspection

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 tools for data scrubbing and data auditing

Data migration and data transforms

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

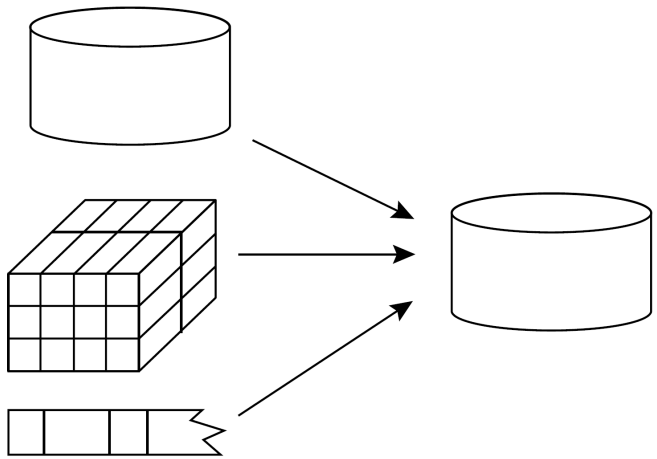
Data Integration

Data Integration

Data Integration

Combining data from multiple sources into a coherent store

- Careful integration can help reduce and avoid redundancies and inconsistencies.
- This can help improve the accuracy and speed of the subsequent data mining process.



Entity identification

Entity identification

Identifying real world entities from multiple data sources.

Detecting and resolving data value conflicts:

- For the same real world entity, attribute values from different sources are different
  e.g. “UK” = “U.K.” = “United Kingdom” = “United Kingdom of Great Britain and Northern Ireland”
- Possible reasons: different representations, different scales
  e.g. temperature in Celsius vs. temperature in Fahrenheit

Handling Redundancy in Data Integration

- Redundant data occur often when integrating multiple databases.
- Derivable data: one attribute may be a “derived” attribute in another table, e.g. annual revenue
- Redundant attributes may be able to be detected by correlation analysis and covariance analysis
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality.

χ^2 Correlation Test for Categorical Data

For categorical data, a **correlation relationship** between two attributes, A and B, can be discovered by a χ^2 (chi-square) test.

- Let (A_i, B_j) denote the joint event that attribute A takes on value a_i and attribute B takes on value b_j, i.e. $(A = a_i, B = b_j)$. A has c distinct values and B has r distinct values.
- The χ^2 value is computed as:

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

where o_{ij} is the observed frequency (i.e. actual count) of the joint event (A_i, B_j) and e_{ij} is the expected frequency of (A_i, B_j).

χ^2 Correlation Test for Categorical Data

- The expected frequency e_{ij} can be computed as:

$$e_{ij} = \frac{count(A = a_i) \cdot count(B = b_j)}{n}$$

  where n is the number of data tuples, $count(A = a_i)$ is the number of tuples having value a_i for A, and $count(B = b_j)$ is the number of tuples having value b_j for B.

- The χ^2 statistic tests the hypothesis that A and B are independent, that is, there is no correlation between them.

χ^2 calculation: an example

	play_chess	not_play_chess	Total
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

Table 1: Contingency table (i.e. observed frequencies) of chess playing versus preferred reading. Numbers in brackets denote expected frequencies.

- **Hypothesis**: chess playing and preferred reading are independent.
- **Chi-square test**:

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

- In a 2x2 table, the χ^2 value needed to reject the hypothesis at the 0.001 significance level is 10.828 $\Rightarrow$ the hypothesis is **rejected**.

Correlation Coefficient for Numeric Data

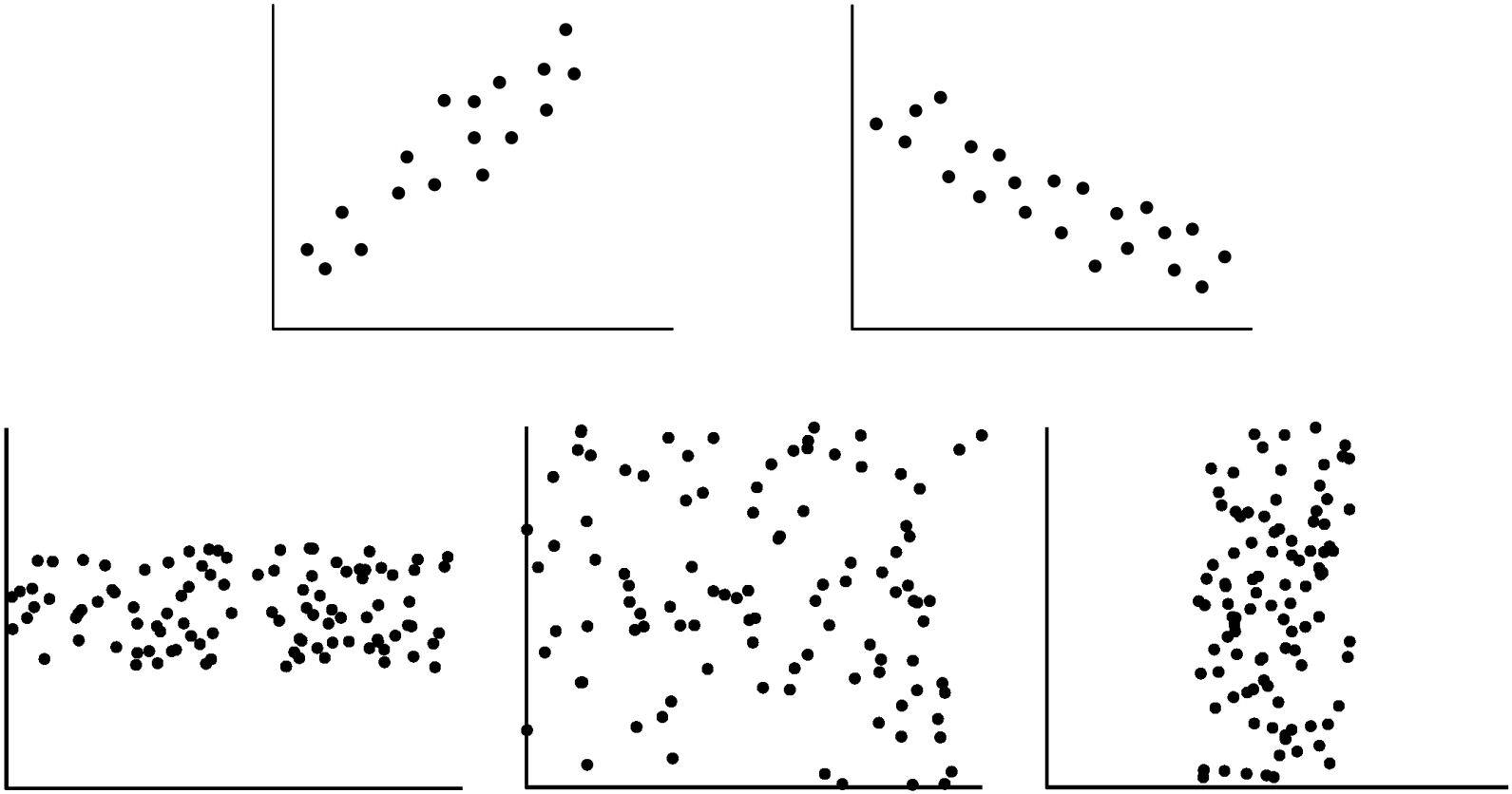
For numeric attributes, we can evaluate the correlation between two attributes, A and B, by computing the correlation coefficient:

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \overline{A})(b_i - \overline{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\overline{AB}}{n\sigma_A\sigma_B}$$

- n is the number of tuples
- a_i and b_i are the respective values of A and B in tuple i
- $\overline{A}$ and $\overline{B}$ are the respective mean values of A and B
- σ_A and σ_B are the respective standard deviations of A and B

Note that $-1 \leq r_{A,B} \leq 1$. If $r_{A,B} > 0$, then A and B are **positively correlated**. If $r_{A,B} = 0$, A and B are **independent**. If $r_{A,B} < 0$, then A and B are **negatively correlated**.

Correlation Coefficient for Numeric Data



Top figures: positively and negatively correlated attributes.
Bottom figures: no observed correlation between attributes.

Data Reduction

Data Reduction

Data Reduction

Obtain a reduced representation of the data set.

- Much smaller in volume but yet produces almost the same analytical results
- **Why data reduction?** Complex analysis may take a very long time to run on a complete data set.
- **Strategies for data reduction**:
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression

Data Reduction: Parametric vs. Non-Parametric Methods

Parametric methods:

- Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
- Example: regression methods (linear regression, log-linear models...)

Non-parametric methods:

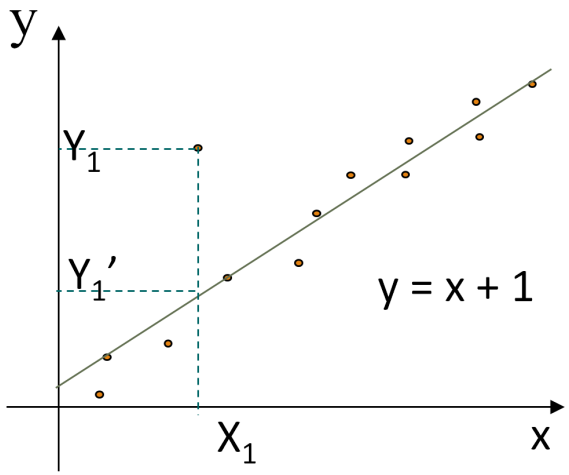
- Do not assume models
- Major families: histograms, clustering, sampling

Data Reduction: Parametric vs. Non-Parametric Methods

Regression analysis

A collective name for techniques for the modeling and analysis of numerical data consisting of values of a dependent variable and of one or more independent variables.

- The parameters are estimated so as to give a “best fit” of the data.
- Most commonly the best fit is evaluated using the least squares method.



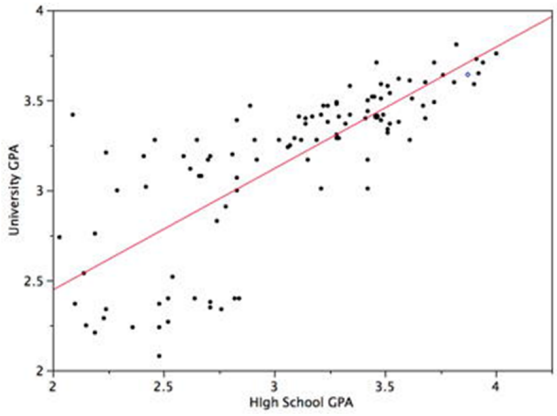
Linear and Multiple Regression

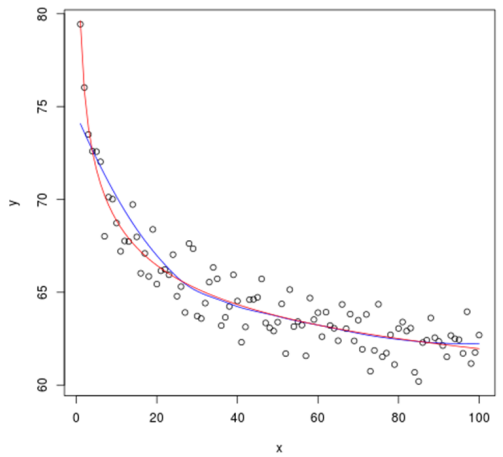
Linear regression: $y = wx + b$

- Data modeled to fit a straight line.
- Often uses the least-squares method.
- Two regression coefficients, w and b, specify the line and are to be estimated using the available data.

Nonlinear regression:

- Data is modeled by a function which is a nonlinear combination of the model parameters and depends on one or more independent variables.





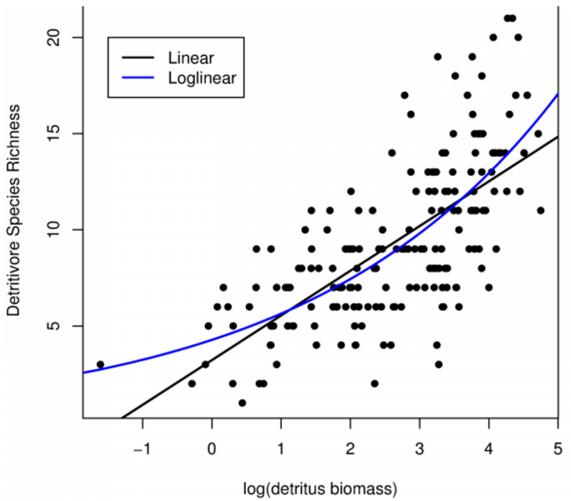
Multiple Regression and Log-Linear Models

Multiple regression: $y = b_0 + b_1 x_1 + b_2 x_2$

- Allows a response variable y to be modeled as a linear function of multidimensional feature vector.
- Many nonlinear functions can be transformed into the above.

Log-linear model:

- Takes the form of a function whose logarithm is a linear combination of the parameters of the model, which makes it possible to apply (possibly multivariate) linear regression.
- Useful for dimensionality reduction and data smoothing



Histograms

A histogram for an attribute *A* partitions the data distribution of *A* into disjoint subsets, referred to as buckets or bins.

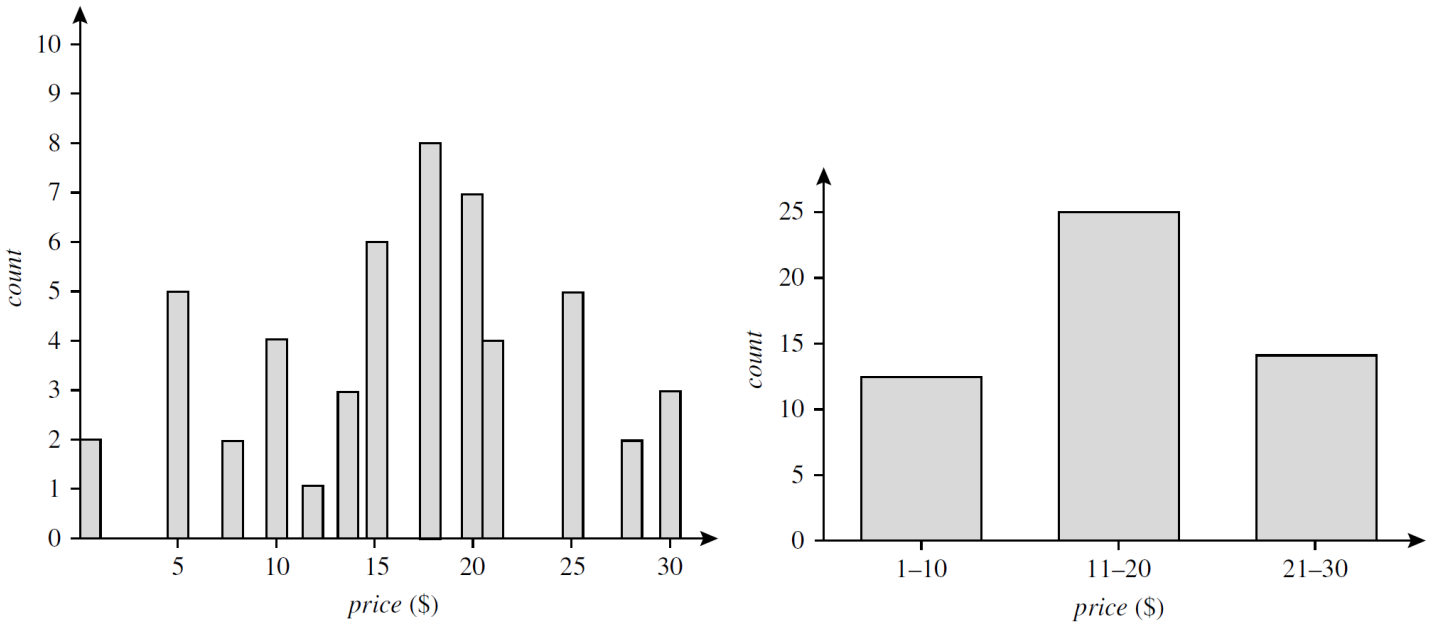
Partitioning rules:

- Equal-width: equal bucket range
- Equal-frequency (or equal-depth)

Histograms are highly effective at approximating both sparse and dense data, as well as highly skewed and uniform data.

Histogram Example

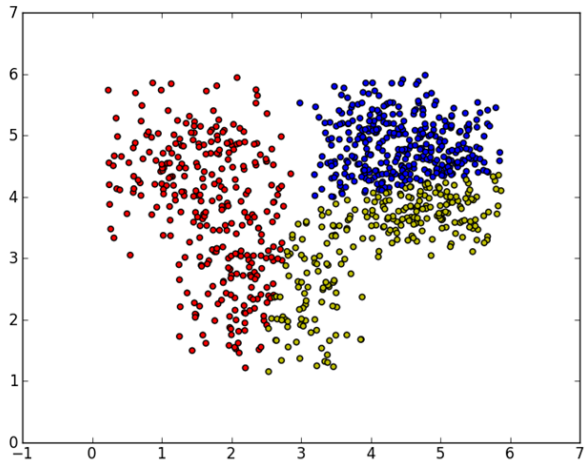
The following data is a list of prices in $ for commonly sold items. The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.



Left: histogram using singleton buckets.
Right: histogram with equal width of $10.

Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g. centroid and diameter) only.
- Can be very effective if data is clustered but less so in other cases.
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures.
- There are many choices of clustering definitions and clustering algorithms (to be studied in week 6).



Sampling

Sampling allows a large data set to be represented by a much smaller and representative data sample (or subset).

Types of sampling:

- Random sampling without replacement: equal probability of selecting any particular item
- Random sampling with replacement: once an object is selected, it is recorded and then replaced (may be drawn again)
- Cluster sampling: if tuples are grouped into clusters, then clusters can be randomly sampled
- Stratified sampling: partition the data set, and draw samples proportionally from each partition

Sampling

T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

Figure: stratified sampling example.

An advantage of sampling for data reduction is that the cost of obtaining a sample is proportional to the size of the sample, as opposed to the size of the dataset size.

Attribute subset selection

Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant or redundant to the mining task.

Redundant attributes:

- Duplicate much or all of the information contained in one or more other attributes.
- e.g. purchase price of a product and the amount of sales tax paid.

Irrelevant attributes:

- Contain no information that is useful for the data mining task at hand.
- e.g. a person's ID number is irrelevant to the task of predicting their income.

Attribute subset selection

Attribute subset selection reduces the data set size by removing irrelevant or redundant attributes (or dimensions).

Goal: find a minimum set of attributes such that the resulting data distribution is close to the original data distribution obtained using all attributes.

How can we find a 'good' subset of the original attributes?

- For n attributes, there are 2^n possible subsets.
- Heuristic methods that explore a reduced search space are commonly used for attribute subset selection.

Attribute subset selection

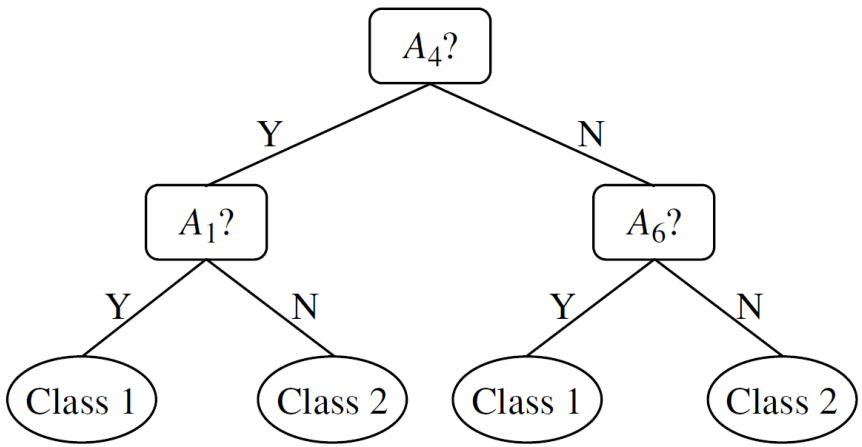
Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$

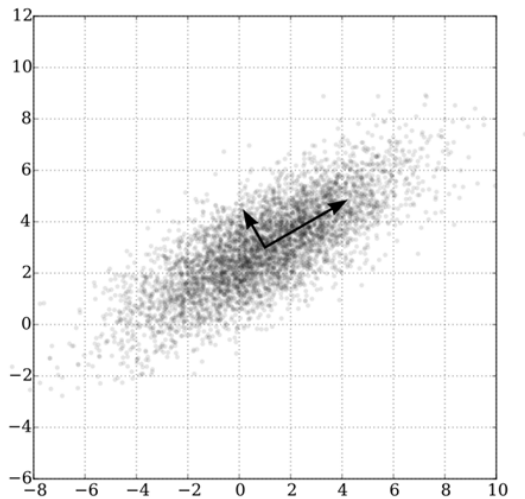
Figure: Heuristic methods for attribute subset selection.

Principal Component Analysis

Principal Component Analysis (PCA)

A statistical procedure that uses a transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

- The original data is projected onto a much smaller space, resulting in dimensionality reduction.
- **Method**: find the eigenvectors of the covariance matrix, and these eigenvectors define the new space.



Principal Component Analysis - Basic Procedure

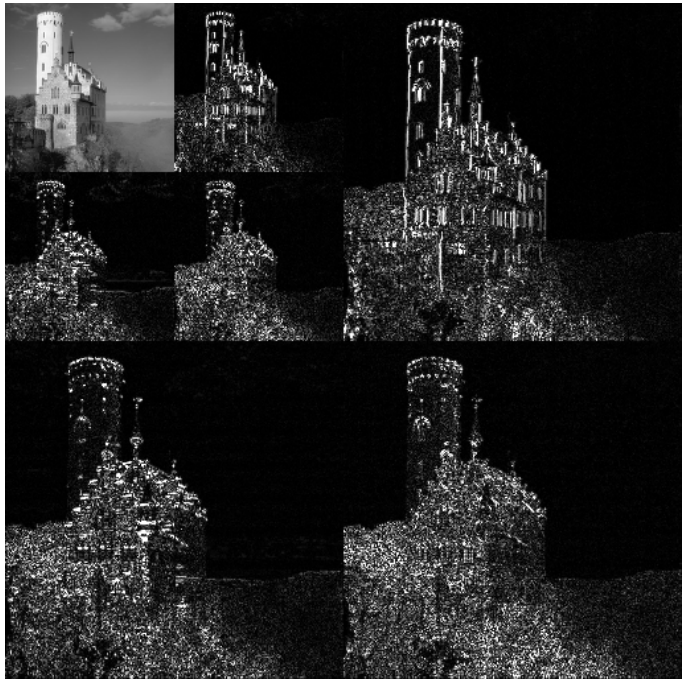
Goal: Given N data vectors from n dimensions, find $k \leq n$ orthogonal vectors (principal components) best used to represent the data.

- Normalise input data using z-score normalisation.
- Compute k orthonormal (unit) vectors, i.e. principal components.
- The input data is a linear combination of the k principal component vectors.
- The principal components are sorted in order of decreasing "significance" or strength.
- Since the components are sorted, the size of the data can be reduced by eliminating the weak components, i.e., those with low variance.

Using the strongest principal components, we can reconstruct a good approximation of the original data.

Wavelet Transform

- Decomposes a signal into a set of wavelets (basis functions) that are orthogonal to translations and scaling
- Applicable to *n*-dimensional signals
- Data are transformed to preserve relative distance between objects at different levels of resolution
- Allows natural clusters to become more distinguishable
- Used for image compression



Wavelet Transform

- Discrete wavelet transform (DWT): signal processing technique that transforms a data vector into wavelet coefficients.
- Compressed approximation: Store only a small fraction of the strongest of the wavelet coefficients
- Similar to discrete Fourier transform (DFT), but better lossy compression.

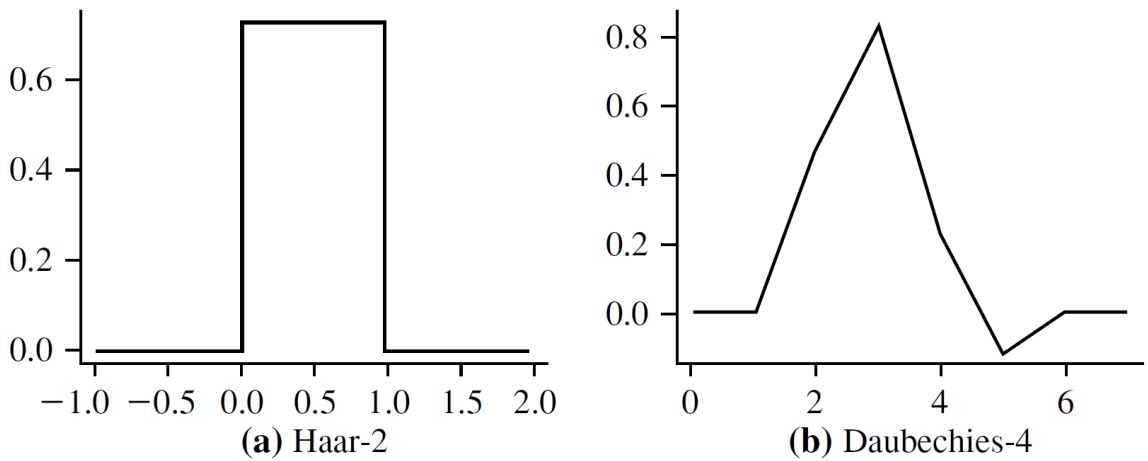


Figure: Examples of wavelet families.

Wavelet Transform

Each transform involves applying two functions recursively. The first applies some data smoothing, such as a sum or weighted average. The second performs a weighted difference, which acts to bring out the detailed features of the data.

Example: data $S = [2, 2, 0, 2, 3, 5, 4, 4]$ can be transformed to $\hat{S} = [2.75, -1.25, 0.5, 0, 0, -1, -1, 0]$.

Resolution	Averages	Detail Coefficients
8	[2, 2, 0, 2, 3, 5, 4, 4]	
4	[2,1,4,4]	[0,-1,-1,0]
2	[1.5,4]	[0.5,0]
1	[2.75]	[-1.25]

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

Data Transformation and Data Discretisation

Data Transformation

Data Transformation

Data are transformed or consolidated so that the resulting mining process may be more efficient, and the patterns found may be easier to understand.

Methods:

- Normalisation: Scaled to fall within a smaller, specified range
  - min-max normalisation
  - z-score normalisation
  - normalisation by decimal scaling
- One-hot encoding
- Discretisation (by binning, clustering, classification)
- Concept hierarchy generation

Normalisation

The measurement unit used in an attribute can affect data analysis. In normalisation, attributes are scaled to fall within a smaller, specified range, such that all attributes have an equal weight in the analysis.

Min-max normalisation: maps a value v of A to v' in the range $[new\_min_A, new\_max_A]$.

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

Example: Normalise the income range £12,000 - £98,000 to [0.0, 1.0]. What is the normalised value for £73,600?

Normalisation

z-score normalisation: the values for an attribute A are normalised based on the mean and standard deviation of A.

$$v' = \frac{v - \mu_A}{\sigma_A}$$

Useful when the actual minimum and maximum of attribute A are unknown, or when there are outliers that dominate the min-max normalisation.

Example: Suppose that the mean and standard deviation of the values for the attribute income are £54,000 and £16,000, respectively. What is the z-score normalised value for £73,600?

Normalisation

Normalisation by decimal scaling: normalises by moving the decimal point of values of attribute A.

$$v' = \frac{v}{10^j}$$

where j is the smallest integer such that $max(|v'|) \leq 1$.

Example: Suppose that the recorded values of A range from -986 to 917. The maximum absolute value of A is 986. To normalise by decimal scaling, we therefore divide each value by 1000 ($j = 3$) so that -986 normalizes to -0.986 and 917 normalises to 0.917.

Note: it is necessary to save the normalisation parameters (e.g. the mean and standard deviation if using z-score normalisation) so that future data can be normalized in a uniform manner.

One-Hot Encoding

Some algorithms can work with categorical data directly. However, many algorithms require all input variables and output variables to be numeric.

One-hot encoding converts categorical data to a numerical form: a new binary variable is added for each unique value.

red	green	blue
1	0	0
0	1	0
0	0	1

Table 2: In the 'colour' variable example, there are 3 categories and therefore 3 binary variables are needed.

Discretisation

Three types of attributes:

- Categorical – values from an unordered set, e.g. colour, profession
- Ordinal – values from an ordered set, e.g. letter grades
- Numeric – real numbers, e.g. integer or real numbers

Discretisation: Divide the range of a continuous attribute into intervals

- Interval labels can then be used to replace actual data values
- Reduce data size by discretisation
- Supervised vs. unsupervised
- Split (top-down) vs. merge (bottom-up)
- Discretisation can be performed recursively on an attribute

Discretisation by Binning

Binning methods were presented previously for data smoothing. These methods are also used as discretisation methods for data reduction.

Binning does not use class information and is therefore an unsupervised discretisation technique. It is sensitive to the user-specified number of bins, as well as the presence of outliers.

Sorted data for *price* (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into (equal-frequency) bins:

Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

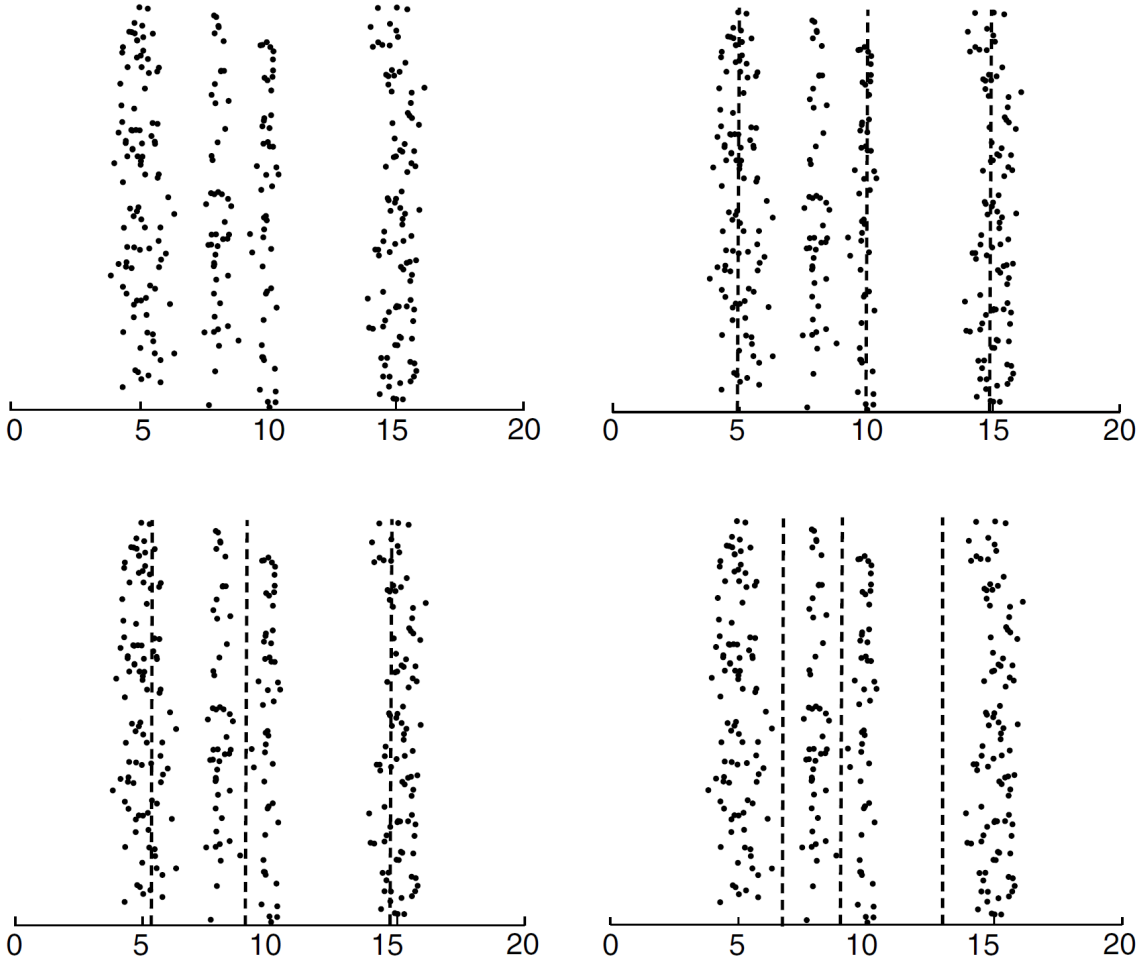
Smoothing by bin means:

Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

Smoothing by bin boundaries:

Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

Discretisation: Binning vs. Clustering



Figures (clockwise starting from top left): data; equal-width binning; clustering; equal-frequency binning.

Discretisation with Supervision

Discretisation using classification

- Make use of class label information: class distribution information is used in the calculation and determination of split points
- Common method: decision trees

Discretisation using correlation analysis

- Also use class information
- Bottom-up approach: finding the best neighboring intervals and then merging them to form larger intervals, recursively.
- Common method: ChiMerge, based on χ^2 test

Concept hierarchy generation

- Concept hierarchy organises concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse.
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity.
- Concept hierarchy formation: recursively reduce the data by collecting and replacing low level concepts (such as numeric values for age) by higher level concepts (such as youth, adult, or senior)
- Concept hierarchies can be either explicitly specified by domain experts or be automatically formed.

Concept Hierarchy Generation for Categorical Data

1. Specification of a partial/total ordering of attributes by users/experts

   - street $<$ city $<$ county

2. Specification of a hierarchy for a set of values by explicit data grouping

   - {Tower Hamlets, Newham, Southwark} $<$ London

3. Specification of a set of attributes, but not of their ordering. The system can then try to automatically generate the attribute ordering.

   - e.g. user specifies street and city as attributes

Automatic Concept Hierarchy Generation

Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set.

- The attribute with the most distinct values is placed at the lowest level of the hierarchy.
- Exceptions, e.g. weekday, month, quarter, year

Summary

Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.

Data integration combines data from multiple sources to form a coherent data store.

Data reduction techniques obtain a reduced representation of the data while minimizing the loss of information content.

Data transformation routines convert the data into appropriate forms for mining.

Data discretisation transforms numeric data by mapping values to interval or concept labels.

Questions?
also please use the forum on QM+