Data Preprocessing

Contents of this Chapter

Introduction

Motivation

- Data mining is based on *existing* data
  different from classical approach in statistics

- Data in the real world is dirty:
  - *incomplete*: lacking certain attributes relevant for the data mining task, lacking attribute values,
  - *noisy*: containing errors or outliers,
  - *inconsistent*: containing discrepancies or contradictions.
- Quality of data mining results crucially depends on quality of input data.

Garbage in, garbage out!

Introduction

Motivation

- Existing data may not be in the structured format required by a data mining algorithm.
- Existing data may not have the required data type.
- Existing data may have
  - too many records, or
  - too many attributes.

Introduction

Data

- Structured data

  records with fixed set of attributes

  attributes have one value of defined type

- Data types

  Nominal (categorical): values from an unordered set

  Boolean: two categorical values

  Ordinal: values from an ordered set

  Continuous (numerical): real numbers

Data

- Unstructured data

  - no record structure

  - complex objects with no attributes and/or with variable number of attributes

→ Document data

→ Image data

→ Time series data

Types of Data Preprocessing

Feature extraction

- Derive meaningful features from the data.

Data cleaning

- Deal with missing values and noisy data.

Data integration

- Integration of multiple datasets, resolution of inconsistencies.

Data transformation

- Normalization, data type conversion.

Data reduction

- Reduction of number of records or attributes.

Feature Extraction

Goal

- Meaningful features are relevant for the given data mining task.
- Meaningful features lead to interpretable results.

→ Feature extraction is an "art" that is highly dependent on the skill of the data scientist.

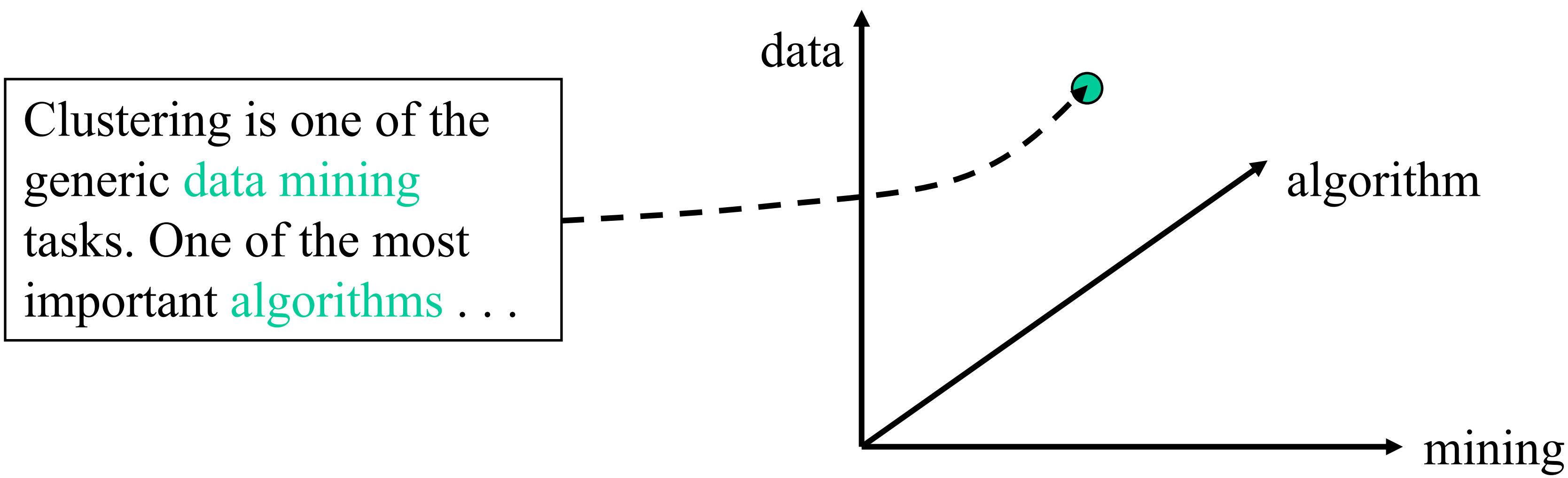
Structured data

- Use attributes as features.
- Derive additional features, where necessary
  e.g. change of profit = profit2017 – profit2016

Feature Extraction

Document data

- Choose relevant terms in document set
  eliminate very rare and very frequent terms,
  use entity recognition to extract terms denoting entities.
- Calculate term frequencies in a document.
- Map document to vector in term space.



Feature Extraction

Image data

- Raw data is represented as matrix of pixels/voxels.
- To extract features, histograms of colors/textures etc. can be used.

Feature Extraction

Image data

•Can partition image into regions and create separate histograms for each region.
•Can employ image segmentation / object recognition to create more semantic features.
•"Visual words" is a semantically rich representation that is similar to document data.
→Feature space tends to be very high-dimensional.

Feature Extraction

Time series data

•Raw data is variable-length sequence of numerical (or categorical) values, which are associated with time stamps.

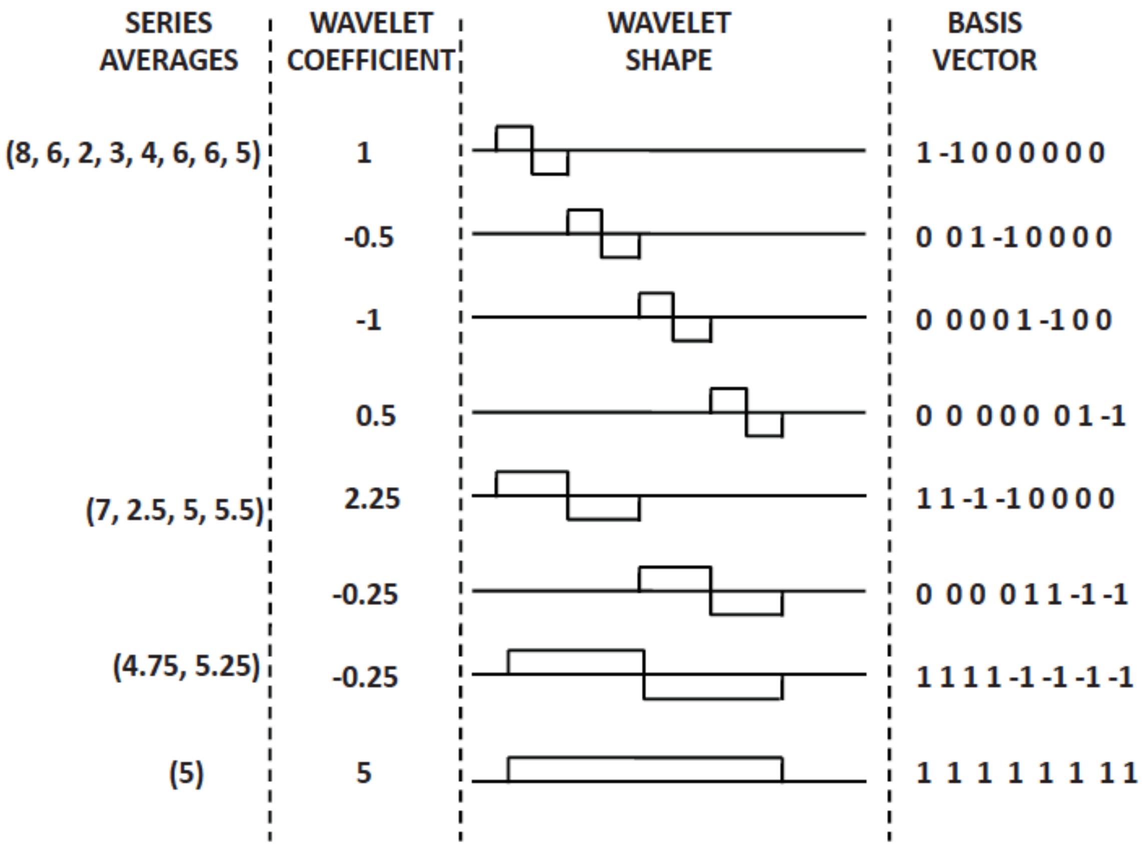
•Naïve approach creates one feature for each pre-defined time window, aggregating all values within the window.

•More sophisticated approaches,
e.g. Discrete Wavelet Transform:

- Representation at multiple levels of resolution
- Averaged differences between different windows
- Subset of the largest coefficients may be used to reduce data size.

Feature Extraction

Discrete Wavelet Transform



Data Transformation

Overview

Normalization

To make different records comparable

→ so that all attributes have similar weights in the data mining process

Convert data types

To allow application of data mining methods for other data type

- Discretization: numerical → ordinal (categorical)
- Binarization: categorical → numerical

Data Transformation

Normalization

Min-max scaling $$v' = \frac{v - \min_a}{\max_a - \min_a}$$

sensitive to outliers

Z-score (standardization) $$v' = \frac{v - \mu_a}{\sigma_a}$$

a : attribute

v : original value

v' : normalized value

μ_a : mean of attribute a

σ_a : standard deviation of attribute a

Data Transformation

Normalization

Percentile rank

•percentage of values that are equal to or lower than *v*

$$v' = \frac{freq(a < v) + 0.5\, freq(a = v)}{N}$$

$freq(a < v)$: number of records with a < v

$freq(a = v)$: number of records with a = v

N : number of all records

Data Transformation

Discretization

Goal

- Reduce the number of values for a given numerical feature by partitioning the range of the feature into intervals.
- Interval labels replace actual feature values.

Methods

- Binning
- Entropy-based discretization

Binning

Equal-width binning

- Divides the range of feature values into N intervals of *equal size.*
- Width of intervals: $Width = \frac{(Max - Min)}{N}$
- Simple.
- Outliers may dominate result.

Equal-depth binning

- Divides the range of feature values into N intervals,
  each containing approximately *same number* of records.
- Outliers and skewed data are also handled well.

Data Transformation

Entropy-Based Discretization

- For classification tasks.
- Given training data set S with class labels $c_1,\ldots,c_k$ and probabilities $p_1,\ldots,p_k$
- Entropy of S $Ent(S) = \sum_{i=1}^{k} -p_i \log p_i$
- If S is partitioned into two intervals S1 and S2 using boundary T,

  the entropy after partitioning is

$$E(S,T) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$

Data Transformation

Entropy-Based Discretization

- Binary discretization: choose boundary that minimizes the entropy function.
- Recursive partitioning of the obtained partitions
  until some stopping criterion is met, e.g.,

$$Ent(S) - E(T, S) \leq \delta$$

Data Transformation

Binarization

• Because binary data is a special form of both numerical and categorical data, it is possible to convert categorical attributes to binary form.

• If a categorical attribute has φ different values, then φ different binary attributes are created, each corresponding to one possible value.

• Exactly one of the φ attributes takes on the value of 1, and the remaining take on the value of 0.

→ Data mining algorithms for numerical data can now be applied.

Data Cleaning

Missing Data

Data is not always available

- E.g., many records have no value for several attributes, such as customer income in sales data.

Missing data may be due to

- Equipment malfunction
- Inconsistent with other recorded data and thus deleted
- Data not entered due to privacy concerns
- Certain data were not considered important at the time of collection
- Data format / contents of database changes in the course of the time

changes with the changing enterprise organization

Data Cleaning

Handling Missing Data

- Ignore the record: usually done when class label is missing.
- Impute missing values

  - Use a default to fill in the missing value:

    e.g., "unknown", a special class, . . .

  - Use the attribute mean or mode to fill in the missing value

    for classification: mean/mode for all records of the same class

  - Use the most probable value to fill in the missing value:

    inference-based such as Bayesian formula or regression

Data Cleaning

Noisy Data

Noise: random error or variance in a measured attribute.

Noisy attribute values may due to

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

Handling Noisy Data

Binning

- Sort data and partition into bins.
- Smooth (i.e., replace data) by bin means, bin median, bin boundaries, etc.

Regression

- Smooth by fitting a regression function.

Clustering

- Detect and remove outliers.

Combined computer and human inspection

- Detect suspicious values automatically and check by human.

Data Cleaning

Binning for Data Smoothing

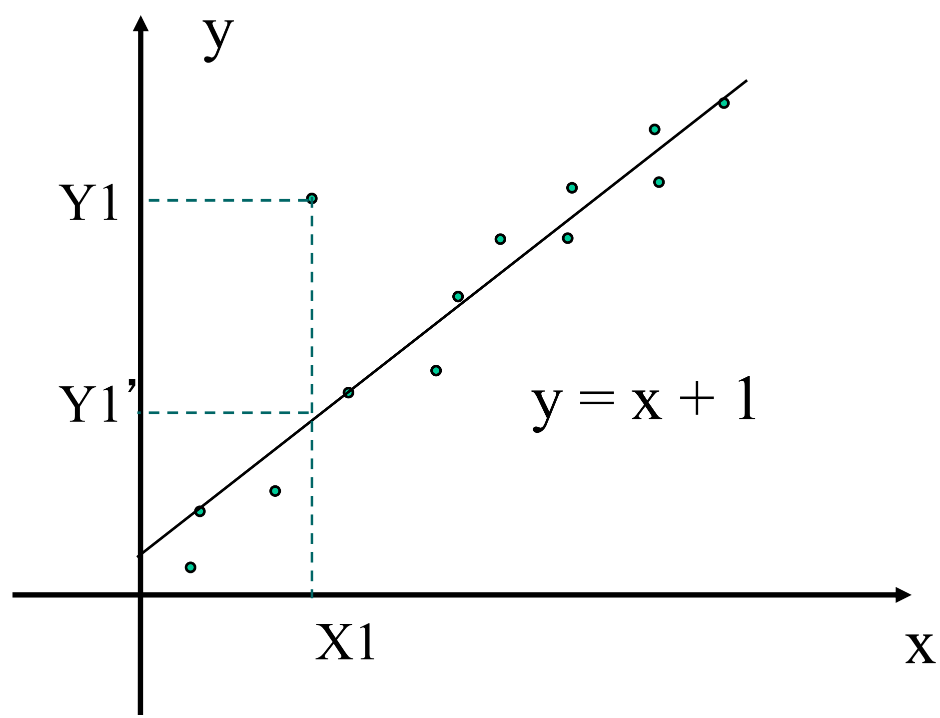
Example: Sorted attribute values 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into three (equi-depth) bins
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
* Smoothing by bin means
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
* Smoothing by bin boundaries
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

Data Cleaning

Regression

- Replace noisy or missing values by predicted values.
- Requires model of feature dependencies (maybe wrong!).
- Can be used for data smoothing or for handling missing data.



Data Integration

Overview

Purpose

- Combine datasets from multiple sources into a coherent dataset (database).

Schema integration

- Integrate metadata from different sources.
- Attribute identification problem: "same" attributes from multiple data sources may have different names.

Instance integration

- Integrate instances from different sources.
- For the same real world entity, attribute values from different sources maybe different.
- Possible reasons:
  different representations, different conventions, different scales, errors.

Data Integration

Approach

Identification

- Detect corresponding tables from different sources
  manual
- Detect corresponding attributes from different sources
  may use correlation analysis
    e.g., A.cust-id ≡ B.cust-#
- Detect duplicate records from different sources
  involves approximate matching of attribute values
    e.g. 3.14283 ≡ 3.1, Schwartz ≡ Schwarz

Treatment

- Merge corresponding tables,
- Use attribute values as synonyms,
- Remove duplicate records.

Data warehouses are already integrated.

Data Reduction

Motivation

Improved efficiency

Runtime of data mining algorithms is typically (super-)linear in the number of records and number of attributes.

Improved quality

Removal of irrelevant attributes and/or records avoids overfitting and improves the quality of the discovered patterns.

→ Reduce number of records and / or number of attributes

Reduced dataset should be representative.

Data Reduction

Feature Selection

Goal

- Select as features the “relevant” subset of the set of all attributes.
- For classification:

  Select a set of features such that the probability distribution of classes given the values for selected attributes is as close as possible to the class distribution given the values of all attributes.

Problem

- 2^d possible subsets of set of d attributes.
- Need heuristic feature selection methods.

Data Reduction

Feature Selection

Feature selection methods

- Feature independence assumption:
  choose features independently by their relevance.
- Greedy bottom-up feature selection:
  - The best single-feature is picked first.
  - Then next best feature conditioned on the first, ...
- Greedy top-down feature elimination:
  - Repeatedly eliminate the worst feature.
- Set-oriented feature selection
  - Consider trade-off between relevance of individual features and the redundancy of feature set.

Data Reduction

Feature Selection

Feature selection criteria

- Mutual information

  - For categorical features

  - measures the information that feature X and class Y share.

- How much does knowing one of the attributes reduce uncertainty about the other?

$$MI(X,Y) = \sum_x \sum_y p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

Data Reduction

Feature Selection

Feature selection criteria

- Fisher score

  - for numerical features and classes $c_1, \ldots, c_k$

  - measures the ratio of the average interclass separation to the average intraclass separation

f1: 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

f2: 0 0 0 0 0 1 1 1 1 1

Data Reduction

Feature Selection

Fisher score

- measures the ratio of the average interclass separation to the average intraclass separation

$$F(f) = \frac{\sum_{i=1}^{k} p_i (\mu_{if} - \mu_f)^2}{\sum_{i=1}^{k} p_i \sigma_{if}^2}$$

p_i : *probability of class i*

μ_{if} : *mean of feature f in class i*

μ_f : *mean of feature f*

σ_{if}^2 : var*iance of feature f in class i*

Data Reduction

Principal Component Analysis (PCA)

Task

- Given *N* data vectors from *d*-dimensional space, find $c << d$ orthogonal vectors that can best represent the data.
- Data representation by projection onto the *c* resulting vectors.
- Best fit: minimal squared error

  error = difference between original and transformed vectors

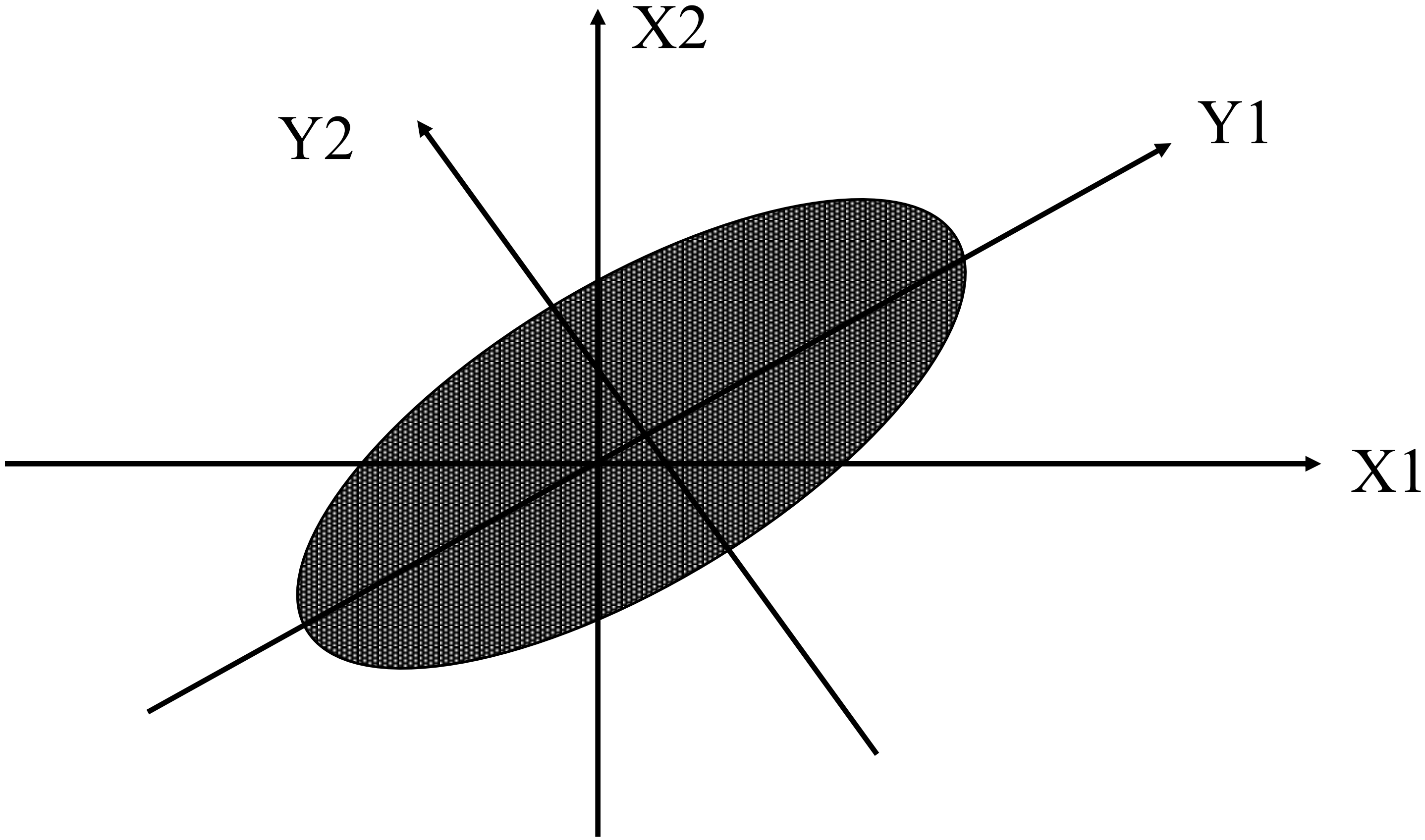
Properties

- Resulting *c* vectors are the directions of the maximum variance of original data.
- These vectors are linear combinations of the original attributes

  maybe hard to interpret!
- Works for numerical data only.

Data Reduction

Example: Principal Component Analysis



Data Reduction

Principal Component Analysis

- $X : n \times d \quad matrix$ representing the training data
  a vector of projection weights (defines resulting vectors)
- $\sigma^2 = (Xa)^T (Xa)$ to be maximized
  $= a^T V a$

  $V = X^T X$ $\quad$ d x d covariance matrix of the training data
- First principal component: eigenvector of the largest eigenvalue of V
- Second principal component: eigenvector of the second largest eigenvalue of V

  and so forth.
- Choose the first k principal components or enough principal components so that the resulting error is below some threshold.

Data Reduction

Sampling

Goal

Choose a *representative* subset of the data records.

Sampling without replacement

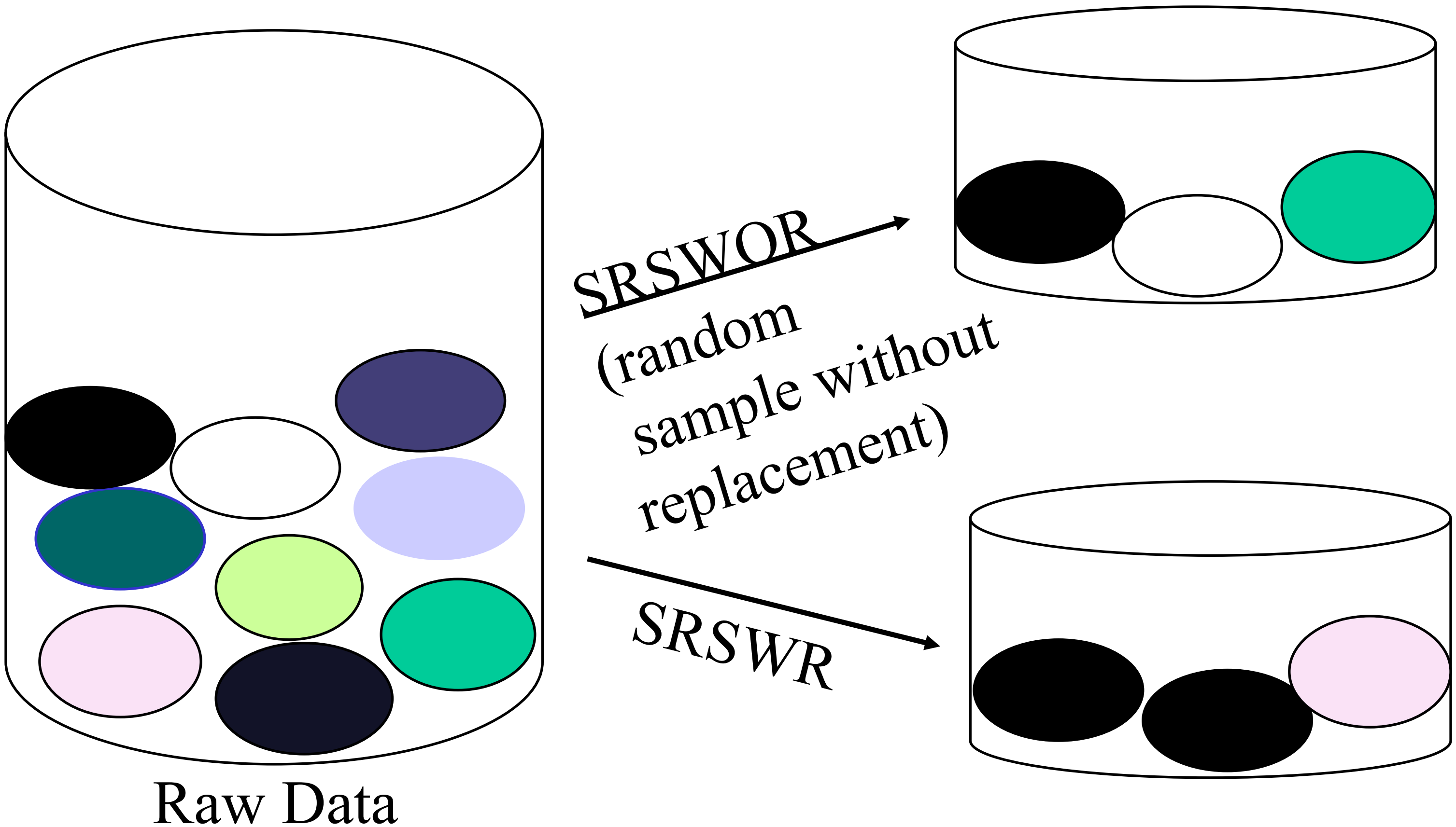
From a data set *D* with *n* records, a total of *n f* records are randomly selected from the not yet selected data.

Sampling with replacement

From a data set D with *n* records, records are sampled independently from the entire data set D for a total of *n f* (possibly duplicate) samples.

Data Reduction

Sampling



Data Reduction

Sampling

Random sampling may overlook small (but important) groups.

Advanced sampling methods

- Biased sampling

  Oversample more important records, e.g. from the minority class.

- Stratified sampling

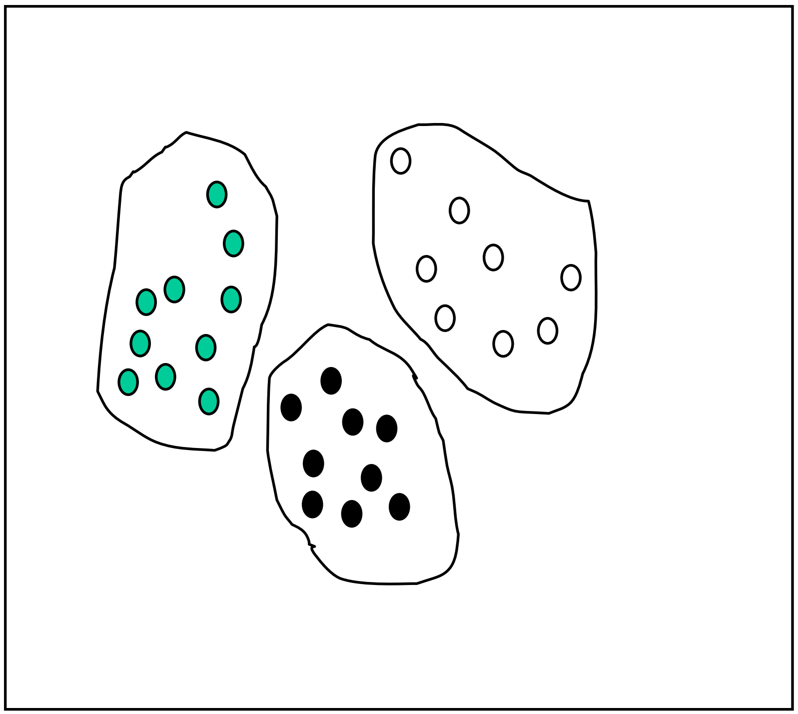
  Draw random samples independently from each given stratum (e.g. age group).

- Cluster sampling

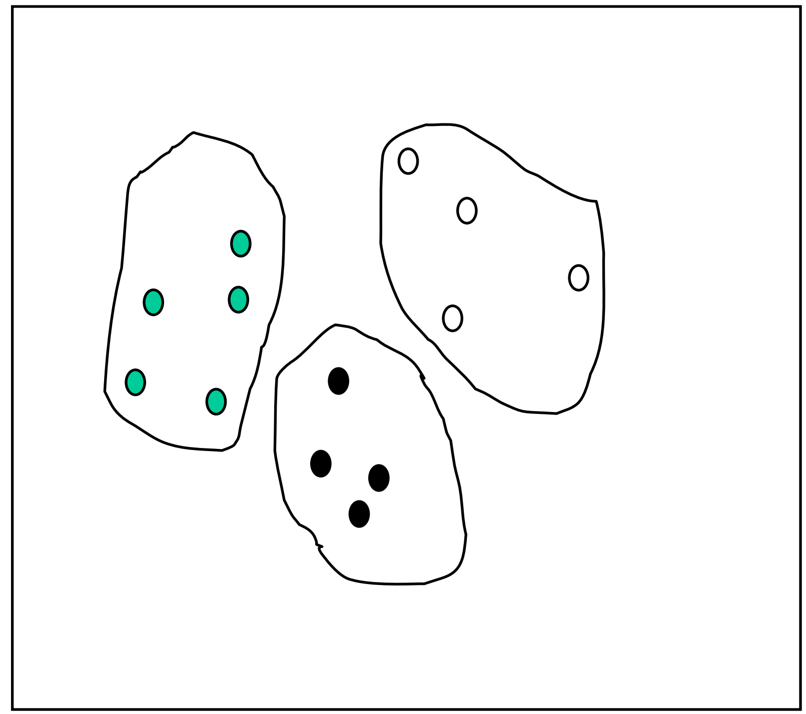
  Draw random samples independently from each given cluster (e.g. customer segment).

Data Reduction

Sampling



Original Data



Cluster/Stratified Sample

One Minute Survey

Assuming that your Data Preprocessing consists of
- feature selection,
- imputing missing data, and
- sampling,

in which order would your perform these three tasks ?