[Data Pre-Processing]

[Rahmad Mahendra, M.Sc.]

Data Mining for Big Data

Pusat Ilmu Komputer Universitas Indonesia

16 – 20 Juli 2018

Course Objectives

- To understand the different problems to solve in the processes of data preprocessing.
- To know the problems in the data integration from different sources and sets of techniques to solve them.
- To know the problems related to clean data and to mitigate imperfect data, together with some techniques to solve them.
- To understand the necessity of applying data transformation techniques.
- To know the data reduction techniques and the necessity of their application.

Agenda

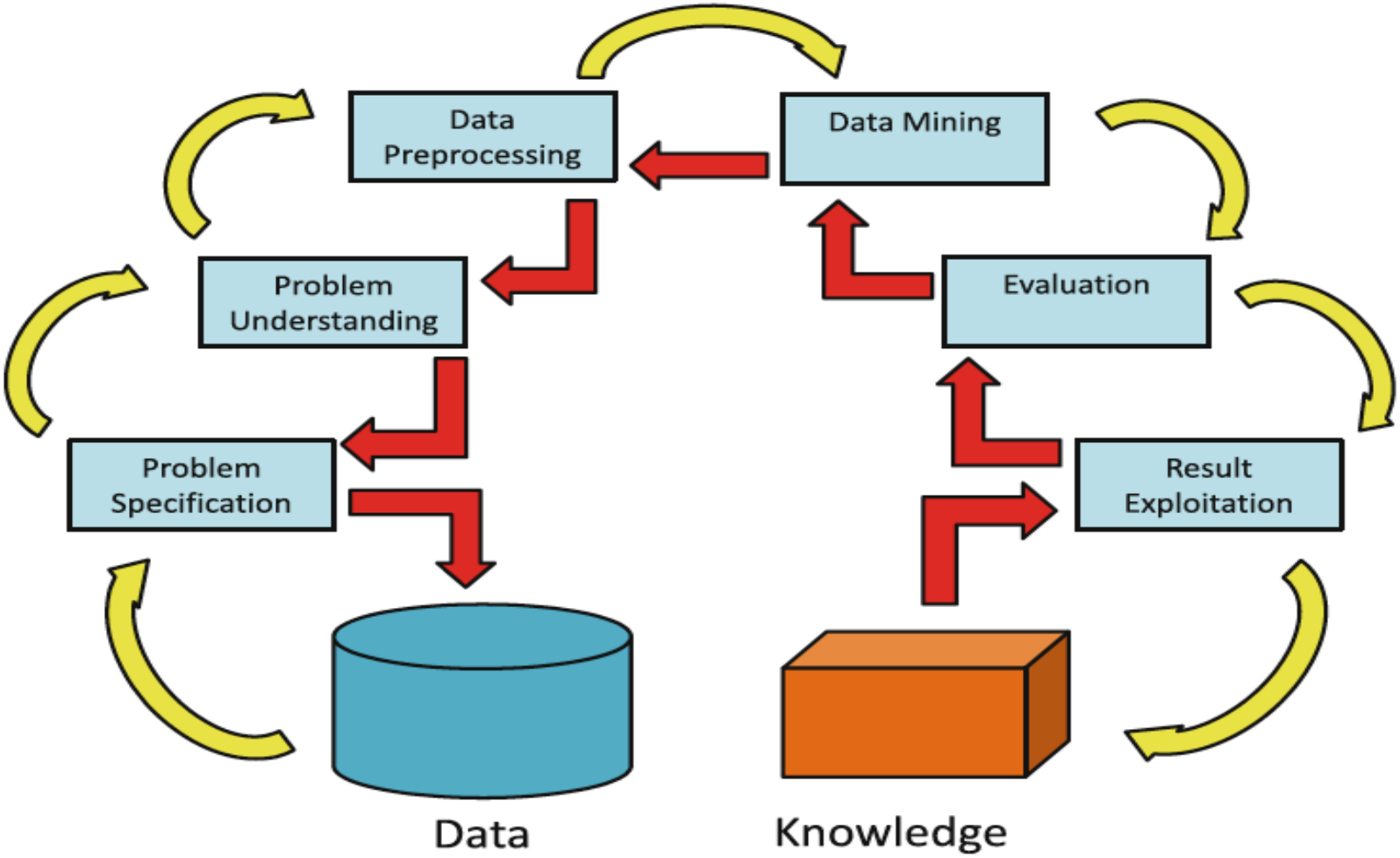
- Introduction to Data Preprocessing
- Data Integration
- Data Cleaning
- Data Transformation
- Data Reduction

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

pusilkom ui

Introduction to Data Preprocessing

Data Mining and Knowledge Discovery



Importance of Data Preprocessing

1. Real data could be dirty and could drive to the extraction of useless patterns/rules.

- This is mainly due to:
  - Incomplete data: lacking attribute values, …
  - Data with noise: containing errors or outliers
  - Inconsistent data (including discrepancies)

Importance of Data Preprocessing

2. Data preprocessing can generate a smaller data set than the original, which allows us to improve the efficiency in the Data Mining process.

- This performing includes Data Reduction techniques: Feature selection, sampling or instance selection, discretization.

Importance of Data Preprocessing

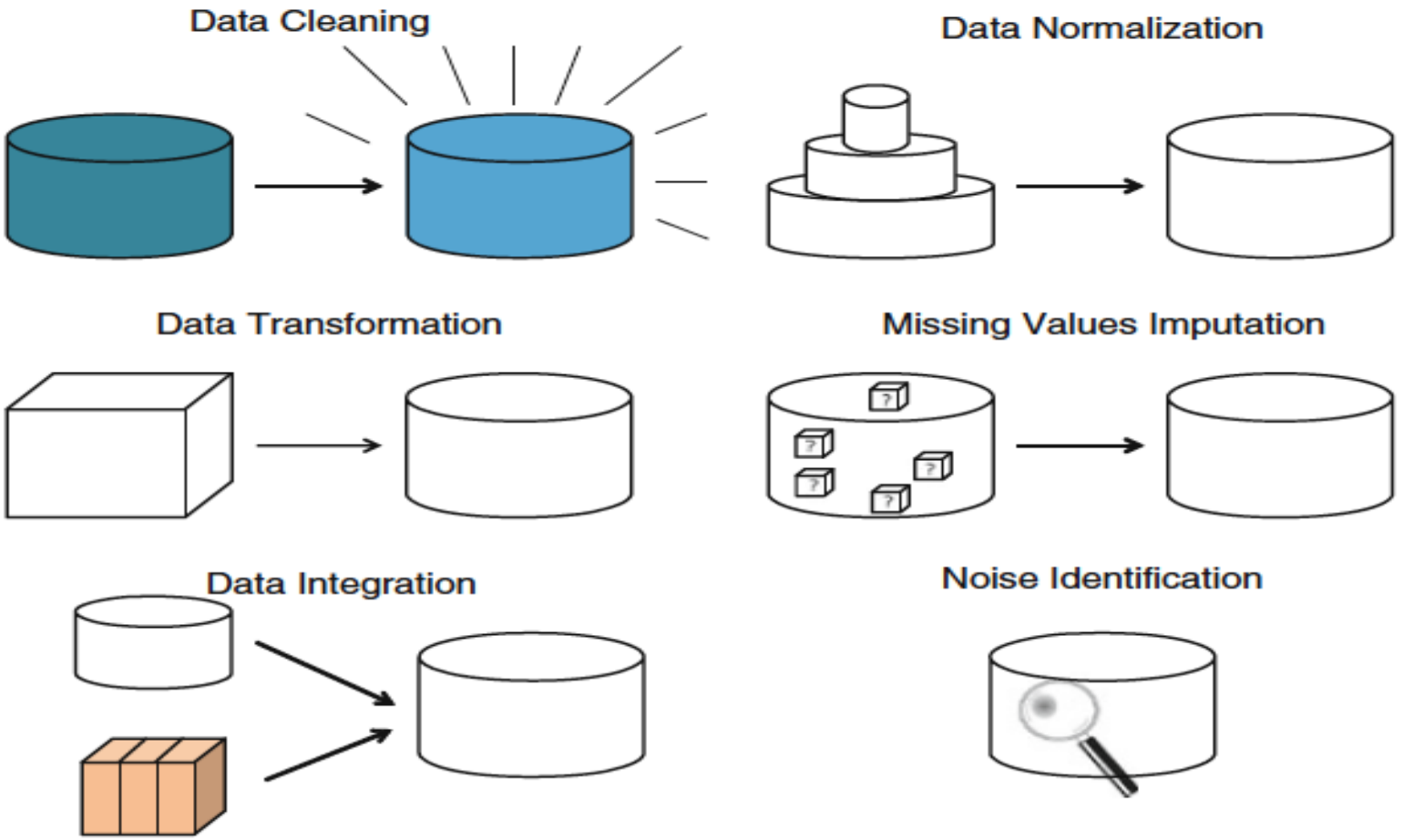
3. No quality data, no quality mining results!

- Data preprocessing techniques generate “quality data”, driving us to obtain “quality patterns/rules”.

Major Tasks in Data Preprocessing

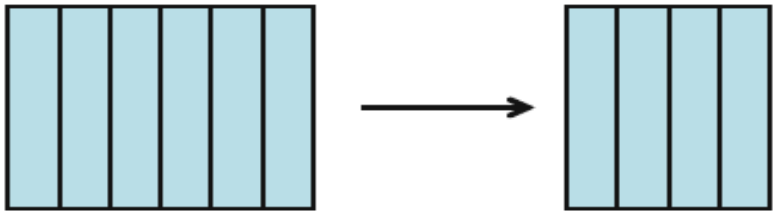
- Data integration.
  - Fusion of multiple sources in a Data Warehousing.
- Data cleaning.
  - Removal of noise and inconsistencies.
- Missing values imputation.
- Data Transformation.
- Data reduction.

Data Preparation Tasks

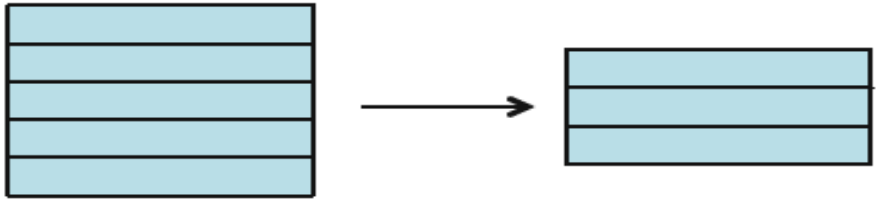


Data Reduction Approaches

Feature Selection



Instance Selection



Discretization

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

pusilkom ui

Data Integration

Data Integration

- Data gathered in data sets can present multiple forms and come from many different sources.
  - Different attribute names or table schemes will produce uneven examples
  - Attribute values may represent the same concept but with different names creating inconsistencies

- Integrating data from different databases is usually called *data integration*.

Data Integration

- **Goal**: collect a single data set with information coming from varied and different sources
- A data map is used to establish how each instance is arranged in a common structure
- Data from relational databases is flattened: gathered together into one single record

Data Integration

Examples

- Different scales: Salary in dollars versus euros (€)

- Derivative attributes: Mensual salary versus annual salary

item	Salary/month
1	5000
2	2400
3	3000

item	Salary
6	50,000
7	100,000
8	40,000

Data Integration

Finding Redundant Attributes

- An attribute is redundant when it can be derived from another attribute or set of them
- Redundancy is a problem that should be avoided
  - It increments the data size → modeling time for DM algorithms increase
  - It also may induce overfitting
- Redundancies in attributes can be detected using correlation analysis

Data Integration

Finding Redundant Attributes

- χ^2 Correlation Test quantifies the correlation among two **nominal** attributes contain *c* and *r* different values each:

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

- where o_{ij} is the frequency of (A_i, B_j) and:

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{m},$$

Data Integration

Finding Redundant Attributes

- χ^2 works fine for nominal attributes, but for numerical attributes Pearson's product moment coefficient is widely

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

- where *m* is the number of instances, and $\overline{A}$,$\overline{B}$ are the mean values of attributes A and B.
- Values of *r* close to +1 or -1 may indicate a high correlation among A and B.

Data Integration

Detecting Tuple Duplication and Inconsistency

- Having duplicate tuples can be a source of inconsistency
- Sometimes the duplicity is subtle
  - If the information comes from different systems of measurement, some instances could be actually the same, but not identified like that
  - Values can be represented using the metric system and the imperial system in different sources

Data Integration

Detecting Tuple Duplication and Inconsistency

- Analyzing the similarity between nominal attributes is not trivial
- Several character-based distance measures for nominal values can be found in the literature:
  - The edit distance
  - The affine gap distance
  - Jaro algorithm
  - q-grams
  - WHIRL distance
  - Metaphone
  - ONCA

Data Integration

Detecting Tuple Duplication and Inconsistency

- Trying to detect similarities in numeric data is harder
- Some authors encode the numbers as strings or use range comparisons ← naïve approaches
- Using the distribution of the data or adapting WHIRL cosine similarity metric are better
- Many authors rely on detecting discrepancies in the data cleaning step

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

pusilkom ui

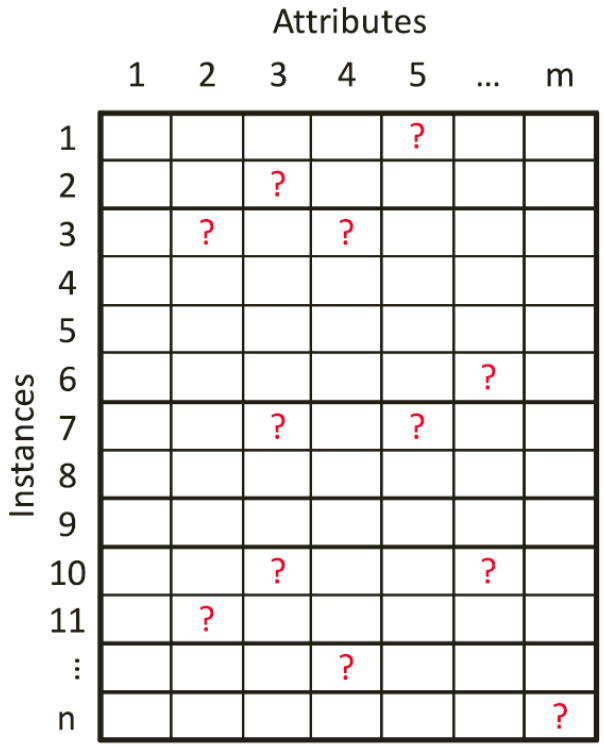
Data Cleaning

Data Cleaning

- Integrating the data in a data set does not mean that the data is free from errors.
- Broadly, dirty data include missing data, wrong data and non-standard representation of the same data.
- If a high proportion of the data is dirty, applying a DM process will surely result in a unreliable model.

Missing Data

- In most problems, the data is arranged in a rectangular data matrix, where MVs can appear as following:



Missing Data

- Data is not always available
  - e.g. many tuples have no recorded values for several attributes
- Missing data may due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not considered important at the time of entry
  - not register history or changes of data

Missing Data

- Missing values make it difficult for analysts to perform data analysis
- Type of problems are associated with missing values (Bernard & Meng, 1999):
  - loss of efficiency;
  - complications in handling and analyzing the data;
  - bias resulting from differences between missing and complete data.

Handling Missing Data

- In general, MVs can be handled in two different ways:
  - Discarding the examples with MVs. Deleting attributes with elevated levels of MVs is included in this category too.
  - Imputation of MVs is a class of procedures that aims to fill in the MVs with estimated ones, as attributes are not independent from each other

Handling Missing Data

- Ignore the tuple: usually done when class label is missing
  - Assuming the task in classification – not effective when the percentage of missing values per attribute varies considerably
- Fill in the missing value manually
  - tedious and infeasible?
- Fill in automatically

Handling Missing Data

- Fill in automatically
  - A global constant, e.g. “unknown”, new class
  - Mean / deviation of the rest of the tuples.
  - Mean / deviation of the rest of the tuples belonging to the same class.
  - Impute with the most probable value. For this, some technique of inference could be used, i.e., Bayesian or decision trees.

Noisy Data

- Noise: random error or variance in measured variable
- Incorrect attribute value may due to:
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems require data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

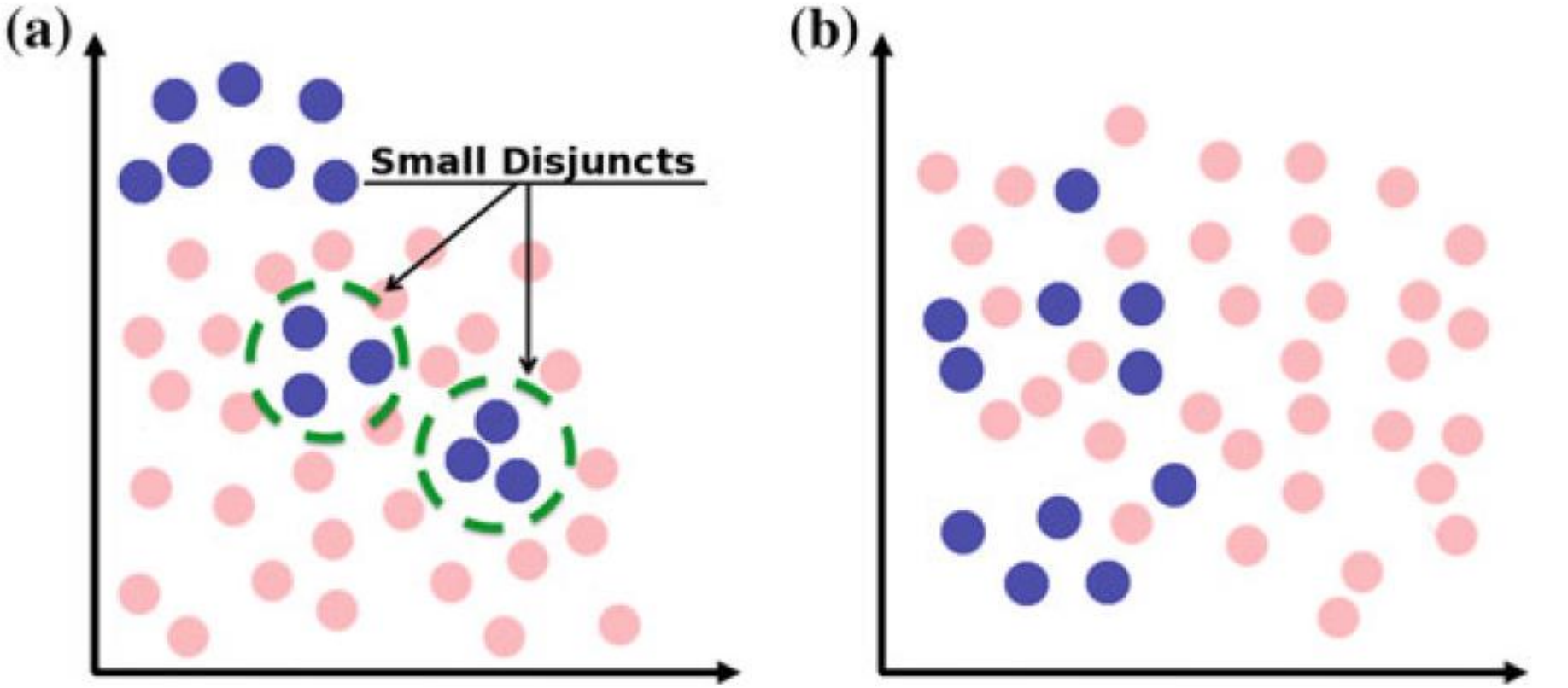
Identifying Noise

- The presence of noise in the data may affect the intrinsic characteristics of a classification problem
  - Noise may create small clusters of instances of a particular class in parts of the instance space corresponding to another class,
  - remove instances located in key areas within a particular class,
  - disrupt the boundaries of the classes and increase overlapping among them.

Identifying Noise

- Noise is not the only problem that supervised ML techniques have to deal with.
- Complex and nonlinear boundaries between classes are problems that may hinder the performance of classifiers
  - it often is hard to distinguish between such overlapping and the presence of noisy examples
- Relevant issues related to the degradation of performance:
  - *Presence of small disjuncts*
  - *Overlapping between classes*

Identifying Noise

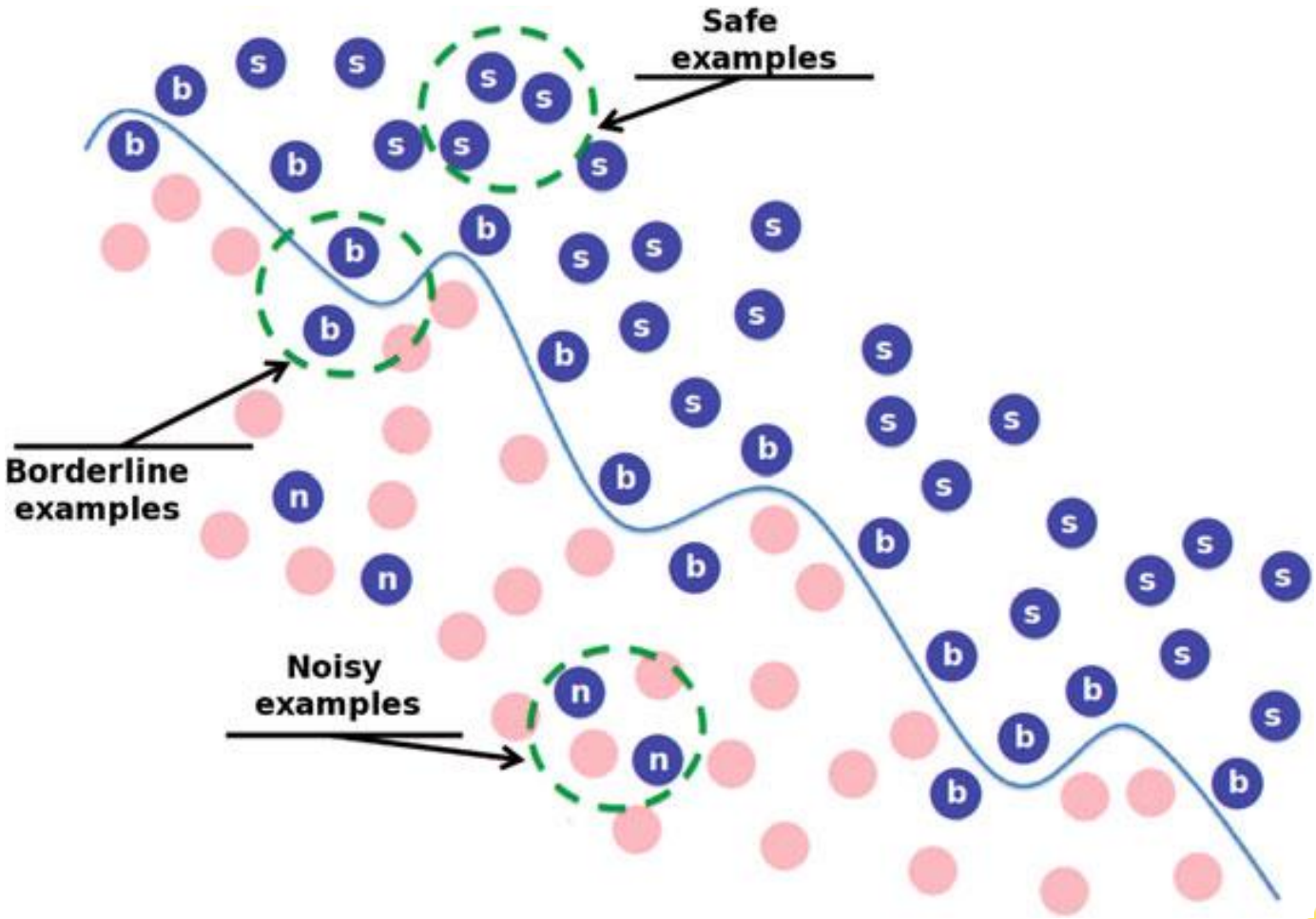


Examples of the interaction between classes: **a** small disjuncts and **b** overlapping

Identifying Noise

- Another interesting problem is pointed out:
  - the higher or lower presence of examples located in the area surrounding class boundaries, which are called borderline examples
- Misclassification often occurs near class boundaries where overlapping usually occurs
- Classifier performance degradation was strongly affected by
  - the quantity of borderline examples
  - the presence of other noisy examples located farther outside the overlapping region

Identifying Noise



Identifying Noise

- *Safe examples* are placed in relatively homogeneous areas with respect to the class label.
- *Borderline examples* are located in the area surrounding class boundaries, where either the minority and majority classes overlap or these examples are very close to the difficult shape of the boundary
- *Noisy examples* are individuals from one class occurring in the safe areas of the other class

Types of Noise

- A large number of components determine the quality of a data set
- Among them, the class labels and the attribute values directly influence such a quality
  - The quality of the class labels refers to whether the class of each example is correctly assigned
  - the quality of the attributes refers to their capability of properly characterizing the examples for classification purposes

Types of Noise

- **Class noise** (also referred as *label noise*) occurs when an example is incorrectly labeled. Two types of class noise can be distinguished:
  - Contradictory examples
  - Misclassifications
- **Attribute noise** refers to corruptions in the values of one or more attributes → erroneous attribute values, missing or unknown attribute values, and incomplete attributes or “do not care” values

Handling Noisy Data

Several approaches have been studied in the literature to deal with noisy data:

- *Robust learners.* These are techniques characterized by being less influenced by noisy data. An example of a robust learner is the C4.5 algorithm
- *Data polishing methods.* Their aim is to **correct** noisy instances prior to training a learner
- *Noise filters* identify noisy instances which can be eliminated from the training data

Handling Noisy Data

- Binning method
  - First, sort data and partition into (equi-depth) bins
  - Then, one can smooth by bin mean, bin median, etc.
- Clustering
  - Detect and remove outliers
- Regression
  - Smoothed by fitting data into regression function
- Combined computer and human inspection
  - Detect suspicious value, checked by human (e.g. deal with possible outliers)

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

pusilkom ui

Data Normalization

Raw Attributes

- Sometimes the attributes selected are *raw attributes.*
  - They have a meaning in the original domain from where they were obtained
  - They are designed to work with the operational system in which they are being currently used

- Usually these original attributes are not good enough to obtain accurate predictive models

Analytic Variables

- It is common to perform a series of manipulation steps to transform the original attributes or to generate new attributes
  - They will show better properties that will help the predictive power of the model

- The new attributes are usually named *modeling variables or analytic variables.*

Data Normalization

Some normalization techniques:

- **Z-score** normalization

$$v' = \frac{v - \overline{A}}{\sigma_A}.$$

- **min-max** normalization: Perform a linear transformation of the original data.

$$[\min_A, \max_A] \rightarrow [new_{\min_A}, new_{\max_A}]$$

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new_{\max_A} - new_{\min_A}) + new_{\min_A}$$

The relationships among original data are maintained.

Data Transformation

Data Transformation

- It is the process to create new attributes
  - Often called transforming the attributes or the attribute set.

- Data transformation usually combines the original raw attributes using different mathematical formulas originated in business models or pure mathematical formulas.

Data Transformation

- Linear transformation
  - Aggregating the information contained in various attributes might be beneficial
  - If *B* is an attribute subset of the complete set *A*, a new attribute *Z* can be obtained by a linear combination:

$$Z = r_1 B_1 + r_2 B2 + \cdots + r_m B_M$$

- Quadratic transformation

$$Z = r_{1,1} B_1^2 + r_{1,2} B_1 B_2 + \cdots + r_{m-1,m} B_{m-1} B_m + r_{m,m} B_m^2,$$

  - where $r_{i,j}$ is a real number.
- Polynomial approximation of transformation
- Non polynomial approximation of transformation

Data Transformation

Box-Cox Transformations

- When selecting the optimal transformation for an attribute is that we do not know in advance which transformation will be the best
- The Box-Cox transformation aims to transform a continuous variable into an almost normal distribution

Data Transformation

Box-Cox Transformations

- This can be achieved by mapping the values using following the set of transformations:

$$y = \begin{cases} x^{\lambda-1}/\lambda, & \lambda \neq 0 \\ log(x), & \lambda = 0 \end{cases}$$

- All linear, inverse, quadratic and similar transformations are special cases of the Box-Cox transformations.

Data Transformation

Box-Cox Transformations

- Please note that all the values of variable *x* in the previous slide must be positive. If we have negative values in the attribute we must add a parameter *c* to offset such negative values:

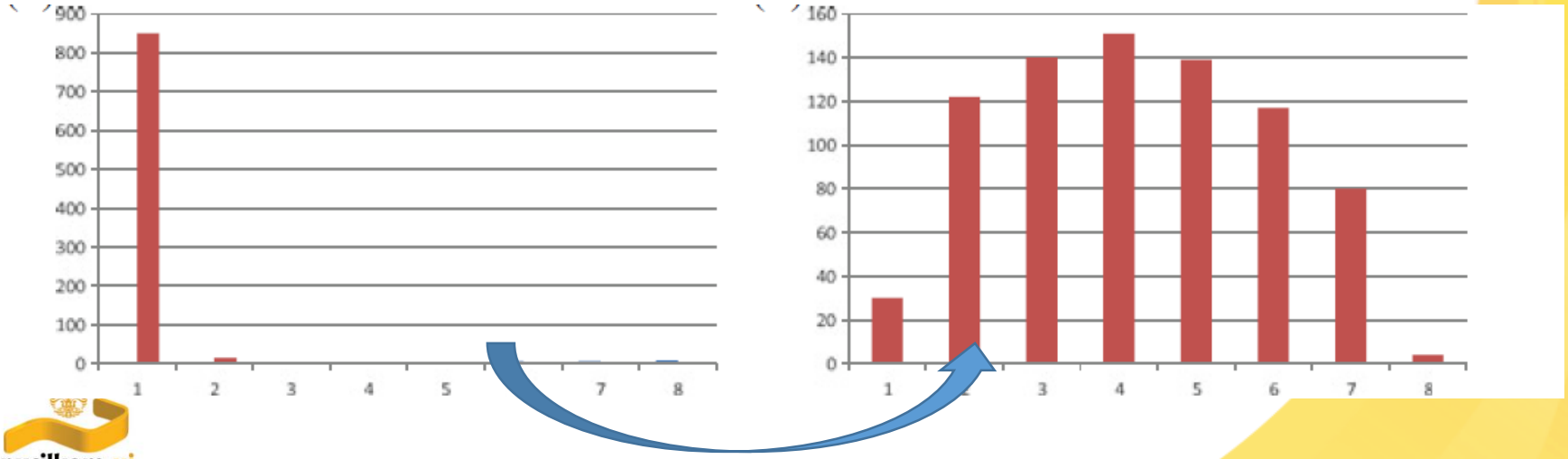
$$y = \begin{cases} (x+c)^{\lambda-1}/g\lambda, & \lambda \neq 0 \\ log(x+c)/g, & \lambda = 0 \end{cases}$$

- The parameter *g* is used to scale the resulting values, and it is often considered as the geometric mean of the data

Data Transformation

Spreading the Histogram

- Spreading the histogram is a special case of Box-Cox transformations
- As Box-Coxtransforms the data to resemble a normal distribution, the histogram is thus spread as shown here



Data Transformation

Spreading the Histogram

- When the user is not interested in converting the distribution to a normal one, but just spreading it, we can use two special cases of Box-Cox transformations
  1. Using the logarithm (with an offset if necessary) can be used to spread the right side of the histogram: $y = log(x)$
  2. If we are interested in spreading the left side of the histogram we can simply use the power transformation $y = x^g$

Data Transformation

Nominal to Binary Transformation

- The presence of nominal attributes in the data set can be problematic, specially if the DM algorithm used cannot correctly handle them
- The first option is to transform the nominal variable to a numeric one
- Although simple, this approach has two big drawbacks that discourage it:
  - With this transformation we assume an ordering of the attribute values
  - The integer values can be used in operations as numbers, whereas the nominal values cannot

Data Transformation

Nominal to Binary Transformation

- In order to avoid the aforementioned problems, a very typical transformation used for DM methods is to map each nominal attribute to a set of newly generated attributes.
- If *N* is the number of different values the nominal attribute has, we will substitute the nominal variable with a new set of binary attributes, each one representing one of the *N* possible values.
- For each instance, only one of the *N* newly created attributes will have a value of 1, while the rest will have the value of 0

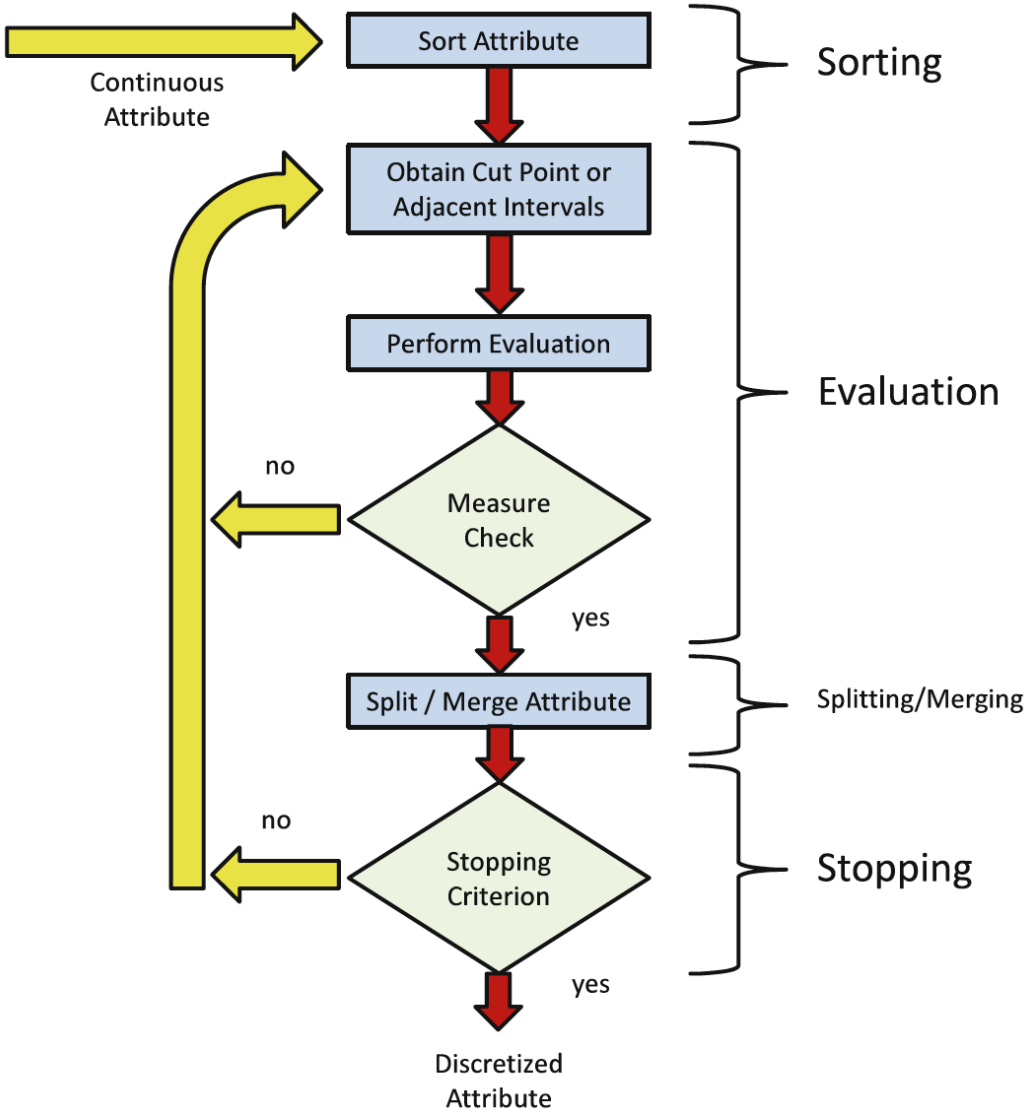
Data Transformation

Nominal to Binary Transformation

- This transformation is also referred in the literature as 1-to-*N transformation.*
- A problem with this kind of transformation appears when the original nominal attribute has a large cardinality
  - The number of attributes generated will be large as well, resulting in a very sparse data set which will lead to numerical and performance problems.

Data Discretization

Data Discretization



Data Discretization

- The discretization is focused on the transformation of continuous values with an order among in nominal/categorical values without ordering. It is also a quantification of numerical attributes.
- Nominal value are within a finite domain, so they are also considered as a data reduction technique.

Data Discretization

- Divide the range of numerical (continuous or not) attributes into intervals.
- Is crucial for association rules and some classification algorithms, which only accepts discrete data.

Age	5	6	6	9	...	15	16	16	17	20	...	24	25	41	50	65	...	67
Owner of a Car	0	0	0	0	...	0	1	0	1	1	...	0	1	1	1	1	...	1

Data Discretization

Advantages

- Many DM algorithms are primarily oriented to handle nominal attributes.
- The reduction and the simplification of data.
- Compact and shorter results.
- Discrete attributes are easier to understand, use, and explain.

Negative effect

- loss of information.

Binning and Reduction of Cardinality

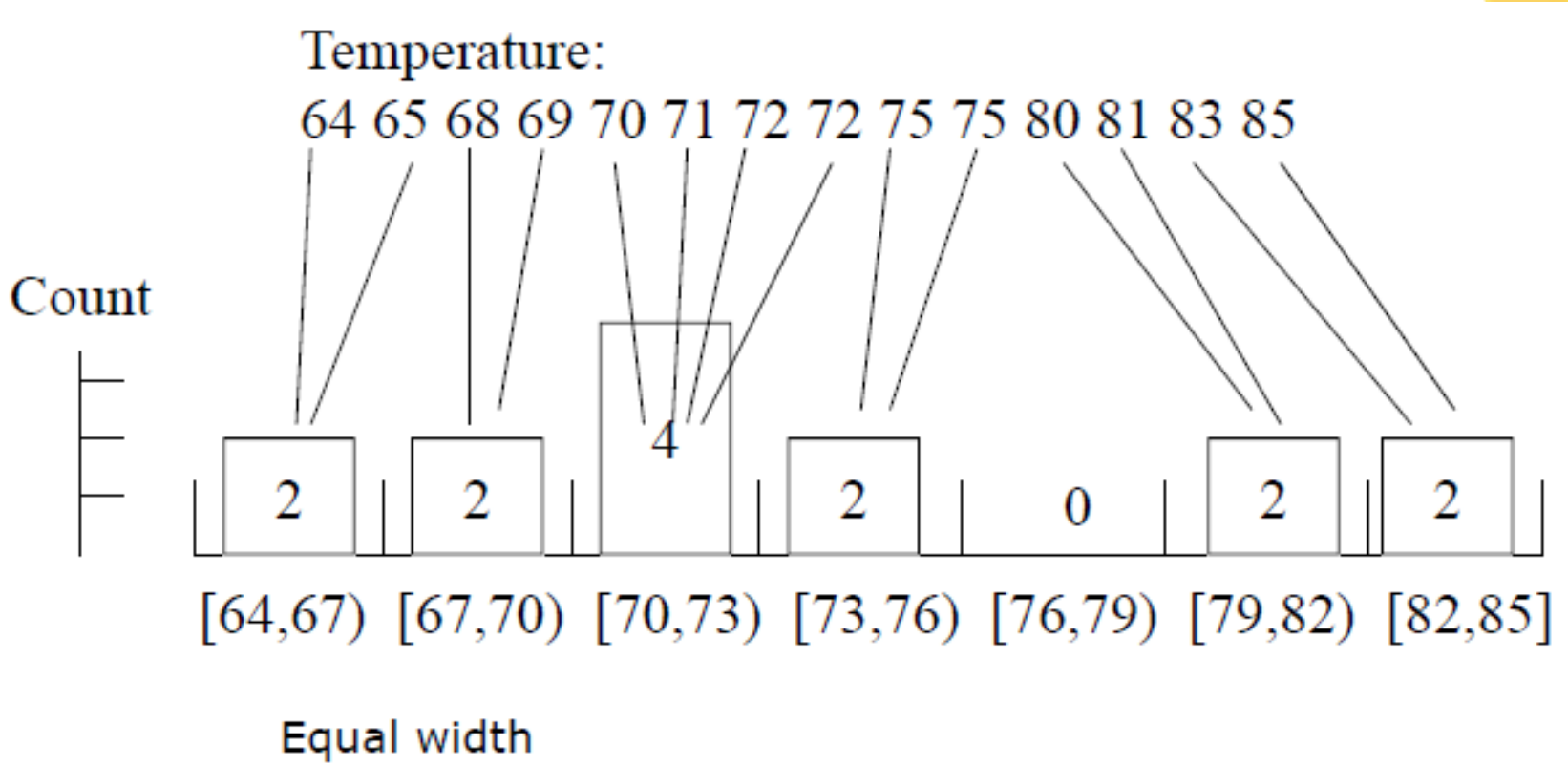
- Binning is the process of converting a continuous variable into a set of ranges.

  (0–5,000; 5,001–10,000; 10,001–15,000, . . . , etc.)

- Cardinality reduction of nominal and ordinal variables is the process of combining two or more categories into one new category.
- Binning is the easiest and direct way of **discretization.**

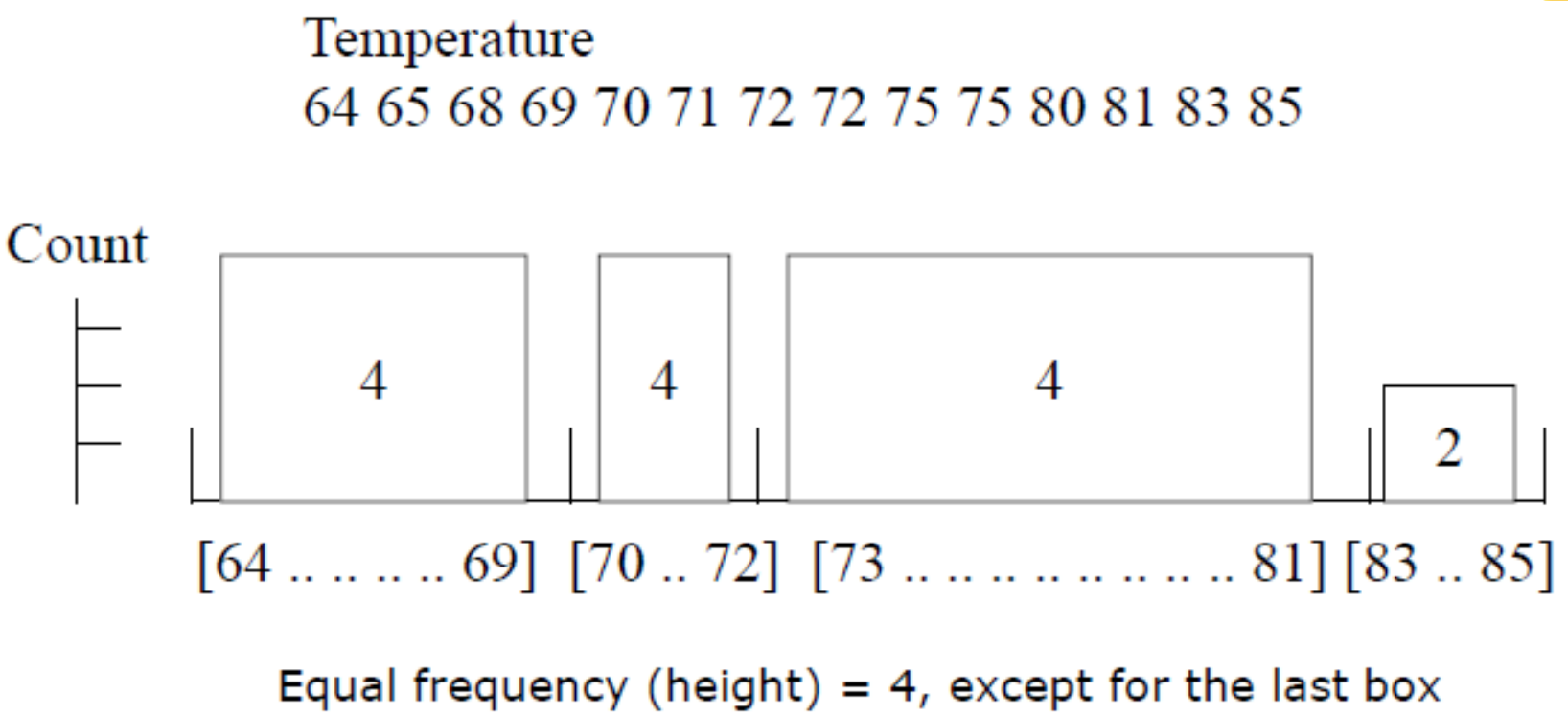
Data Discretization

- Example: equal width



Data Discretization

- Example: equal frequency



Equal frequency (height) = 4, except for the last box

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

pusilkom ui

Data Reduction

Data Reduction

- When the data set is very large, performing complex analysis and DM can take a long computing time
- **Data reduction techniques** are applied in these domains to reduce the size of the data set while trying to maintain the integrity and the information of the original data set as much as possible
- Mining on the reduced data set will be much more efficient and it will also resemble the results that would have been obtained using the original data set.

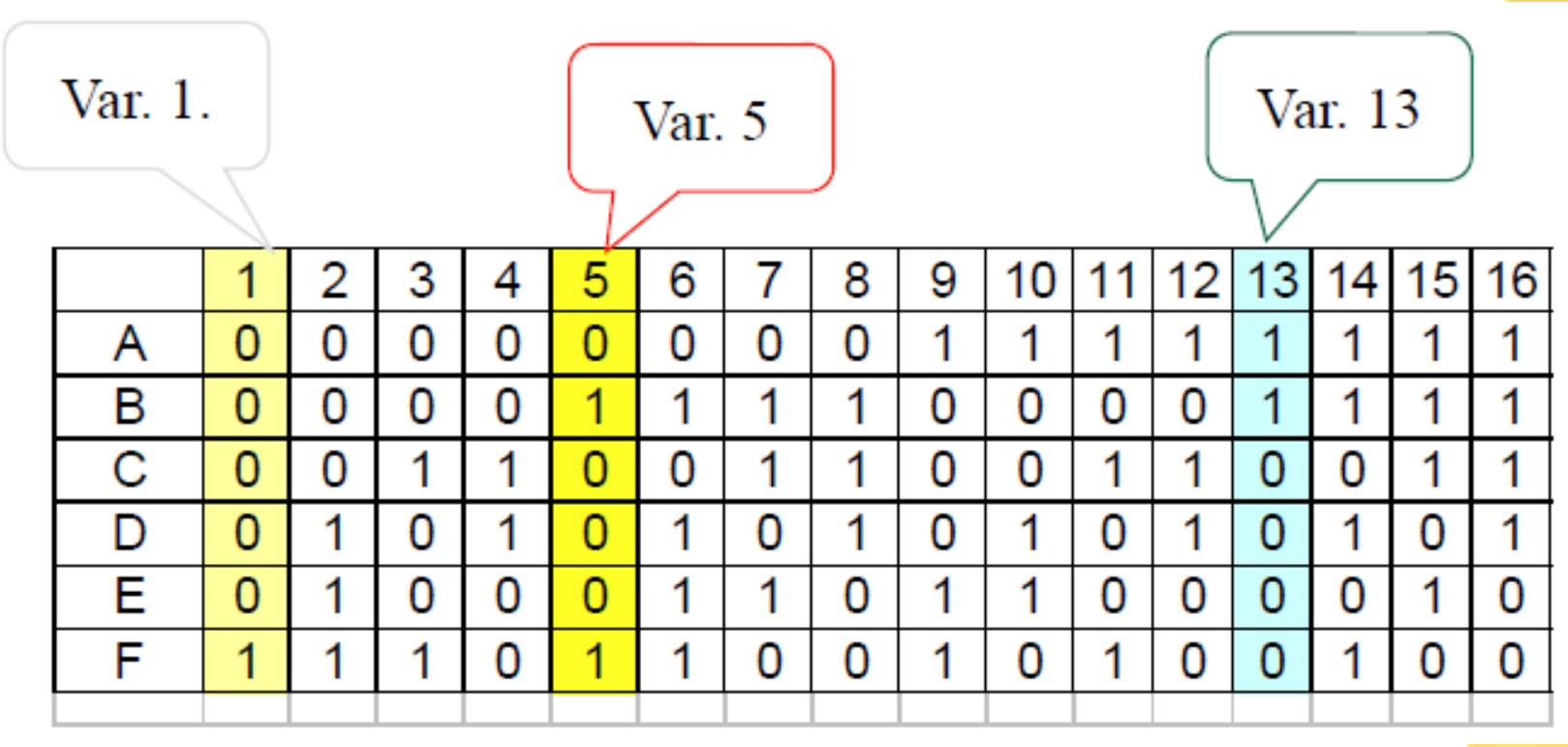
Data Reduction

- **Dimensionality Reduction:** ensures the reduction of the number of attributes or random variables in the data set.
- **Sample Numerosity Reduction:** replaces the original data by an alternative smaller data representation
- **Cardinality Reduction:** transformations applied to obtain a reduced representation of the original data

Curse of Dimensionality

- Dimensionality becomes a serious obstacle for the efficiency of most of the DM algorithms.
- It has been estimated that as the number of dimensions increase, the sample size needs to increase exponentially in order to have an effective estimate of multivariate densities.
- Feature Selection methods are aimed at eliminating irrelevant and redundant features, reducing the number of variables in the model.

Feature Selection



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
B	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
C	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
E	0	1	0	0	0	1	1	0	1	1	0	0	0	0	1	0
F	1	1	1	0	1	1	0	0	1	0	1	0	0	1	0	0

Feature Selection

- *Feature Selection* is a process that chooses an optimal subset of features according to a certain criterion.

- Why we need Feature Selection:
  - to improve performance (in terms of speed, predictive power, simplicity of the model).
  - to visualize the data for model selection.
  - to reduce dimensionality and remove noise.

Feature Selection

Reasons for performing Feature Selection may include:

- removing irrelevant data.
- increasing predictive accuracy of learned models.
- reducing the cost of the data.
- improving learning efficiency, such as reducing storage requirements and computational cost.
- reducing the complexity of the resulting model description, improving the understanding of the data and the model.

Feature Selection

- The outcome of Feature Selection would be:
  - Less data → algorithms could learn quickly
  - Higher accuracy → the algorithm better generalizes
  - Simpler results → easier to understand them

- Feature Selection has as extension the extraction and construction of attributes.

Feature Selection

- **Filter.** The goal function evaluates the subsets basing on the information they contain.
  - Measures of class separability, statistical dependences, information theory are used as the goal function.
- **Wrapper.** The goal function consists of applying the same learning technique that will be used later over the data resulted from the selection of the features.
  - The returned value usually is the accuracy rate of the constructed classifier.

Feature Selection

Advantages

- **Wrappers:**
  - **Accuracy:** generally, they are more accurate than filters, due to the interaction between the classifier used in the goal function and the training data set.
  - **Generalization capability:** they pose capacity to avoid overfitting due to validation techniques employed.
- **Filters:**
  - **Fast:** They usually compute frequencies, much quicker than training a classifier.
  - **Generality:** Due to they evaluate intrinsic properties of the data and not their interaction with a classifier, they can be used in any problem.

Feature Selection

Drawbacks

- **Wrappers:**
  - **Very costly:** for each evaluation, it is required to learn and validate a model. It is prohibitive to complex classifiers.
  - **Ad-hoc solutions:** The solutions are skewed towards the used classifier.
- **Filters:**
  - **Trend to include many variables:** Normally, it is due to the fact that there are monotone features in the goal function used. The use should set the threshold to stop.

Data Sampling

- To reduce the number of instances submitted to the DM algorithm.
- To support the selection of only those cases in which the response is relatively homogeneous.
- To assist regarding the balance of data and occurrence of rare events.
- To divide a data set into three data sets to carry out the subsequent analysis of DM algorithms.

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

pusilkom ui

Summary

Data Preprocessing

- Data Integration
  - Merging of data from multiple data stores.
- Data Cleaning
  - Correct bad data, filter some incorrect data out of the data set and reduce the unnecessary detail of data.
- Data Transformation
  - The data is consolidated so that the mining process result could be applied or may be more efficient.

Data Preprocessing

- Data Normalization
  - To express data in the same measurements units, scale or range.
- Missing Data Imputation
  - To fill the variables that contain missing values with some intuitive data.
- Noise Identification
  - To detect random errors or variances in a measured variable.

Data Preprocessing

- Feature Selection
  - Achieves the reduction of the data set by removing irrelevant or redundant features (or dimensions).
- Instance Selection
  - Consists of choosing a subset of the total available data to achieve the original purpose of the DM application as if the whole data had been used.

Data Preprocessing

- Discretization
  - Transforms quantitative data into qualitative data, that is, numerical attributes into nominal attributes with a finite number of intervals.
- Feature Extraction / Instance Generation
  - Extends both the feature and instance selection by allowing the modification of the internal values that represent each example or attribute.

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

pusilkom ui

Thank You