



Data Pre-Processing

Data Pre-Processing

Data Objects and Attribute Types

Measuring Data Similarity and Dissimilarity,

Why Pre-process the Data

Data Cleaning

Data Integration

Data Reduction

Data Transformation

Data Discretization.

DISCRETE AND CONTINUOUS

Discrete

- A discrete attribute has a finite or countably infinite set of values.
- Such attributes can be categorical, such as zip codes or ID numbers, or numeric, such as counts.
- Discrete attributes are often represented using integer variables.

Binary

- Binary attributes are a special case of discrete attributes and assume only two values, e.g., true/false, yes/no, male/female, or 0/1.
- Binary attributes are often represented as Boolean variables, or as integer variables that only take the values 0 or 1.

Continuous

- A continuous attribute is one whose values are real numbers.
- Examples include attributes such as temperature, height, or weight.
- Continuous attributes are typically represented as floating-point variables.
- Practically, real values can be measured and represented only with limited precision.

Major Tasks in Data Preprocessing



Data cleaning

Fill in missing values, 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 compression



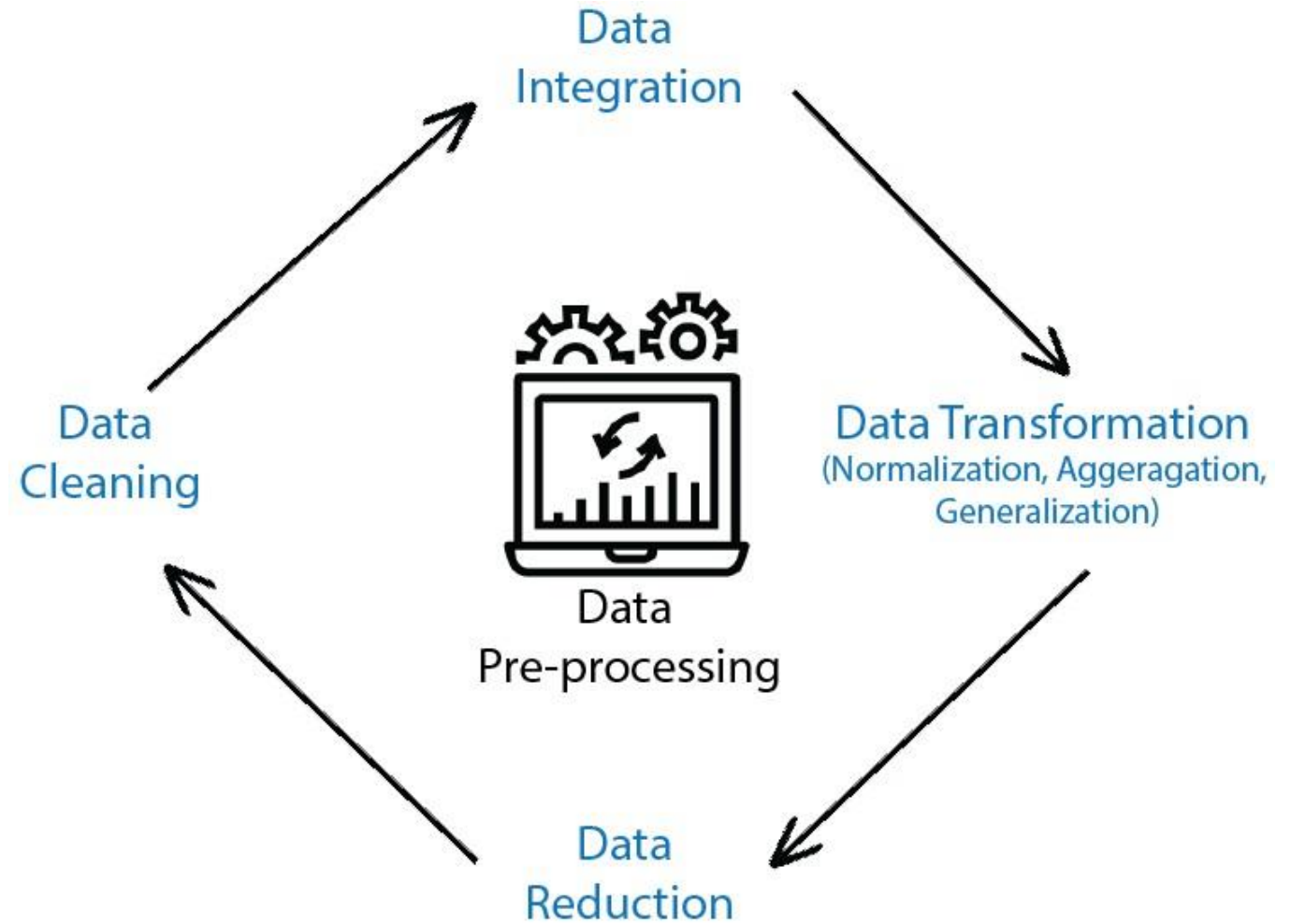
Data transformation and data discretization

Normalization

Concept hierarchy generation

Data Preprocessing

- Potential issues with data
- E.g., missing data, errors, inconsistency, availability
- Preparing data for the mining process
- Data cleaning, integration, transformation, reduction
- No good data, no good data mining!



Data Quality



Data Quality: Why Preprocess the Data?

- 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, ...
 - Timeliness: timely update?
 - Believability: how trustable the data are correct?
 - Interpretability: how easily the data can be understood?

Issues in Real-world Data

Incomplete

Missing values,
missing
attributes

Noisy

Imprecision,
errors, outliers:
e.g., age = “-10”

Inconsistent

E.g., age vs.
birthday, rating
scale

Causes of Data Issues



Data collection/transmission/processing



Human, hardware, and software



Limitations, errors, multiple sources



Changes over time



Updated survey, new sensing capabilities



Data Cleaning

Data in the Real World Is Dirty:

- Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error

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 codes or names, e.g.,

- *Age*="42", *Birthday*="03/07/2010"
- Was rating "1, 2, 3", now rating "A, B, C"
- discrepancy between duplicate records

Intentional (e.g., *disguised missing data*)

- Jan. 1 as everyone's birthday?

Data Mining: Preprocessing Techniques

Data Quality

Follow Discussions of Ch. 2 of the Textbook

- Aggregation
- Sampling
- Dimensionality Reduction
- Feature subset selection
- Feature creation
- Discretization and Binarization
- Attribute Transformation

Similarity Assessment (part of the clustering transparencies)

Incomplete (Missing) Data

Data is not always available

- E.g., many tuples have no recorded 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 misunderstanding
- certain data may not be considered important at the time of entry
- 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* (when doing classification)—not effective when the % of missing values per attribute varies considerably

Fill in the missing value manually: tedious + infeasible?

Fill in it automatically with

- a global constant : e.g., “unknown”, a new class?!
- the attribute mean
- the attribute mean for all samples belonging to the same class: smarter
- the most probable value: inference-based such as *Bayesian formula or decision tree*

Noisy Data

Noise: random error or variance in a measured variable

Incorrect attribute values may be due to

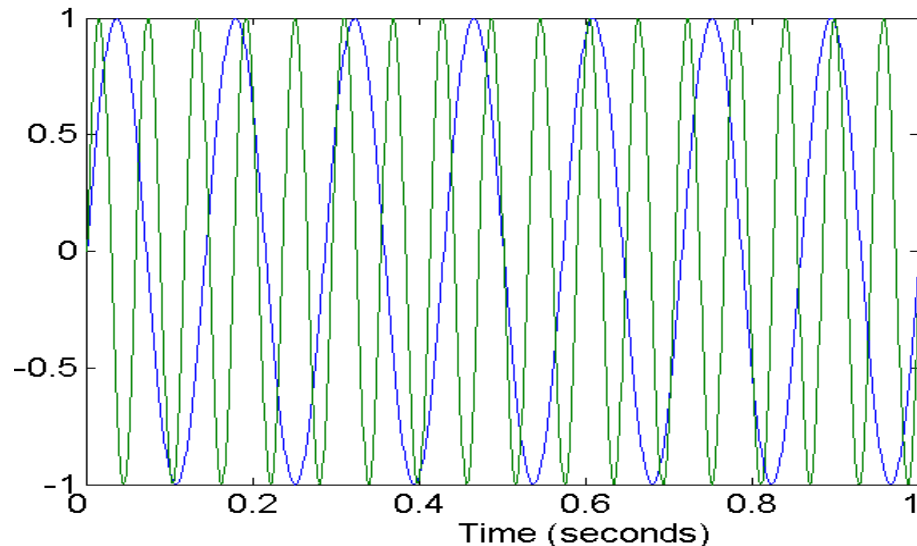
- faulty data collection instruments
- data entry problems
- data transmission problems
- technology limitation
- inconsistency in naming convention

Other data problems which require data cleaning

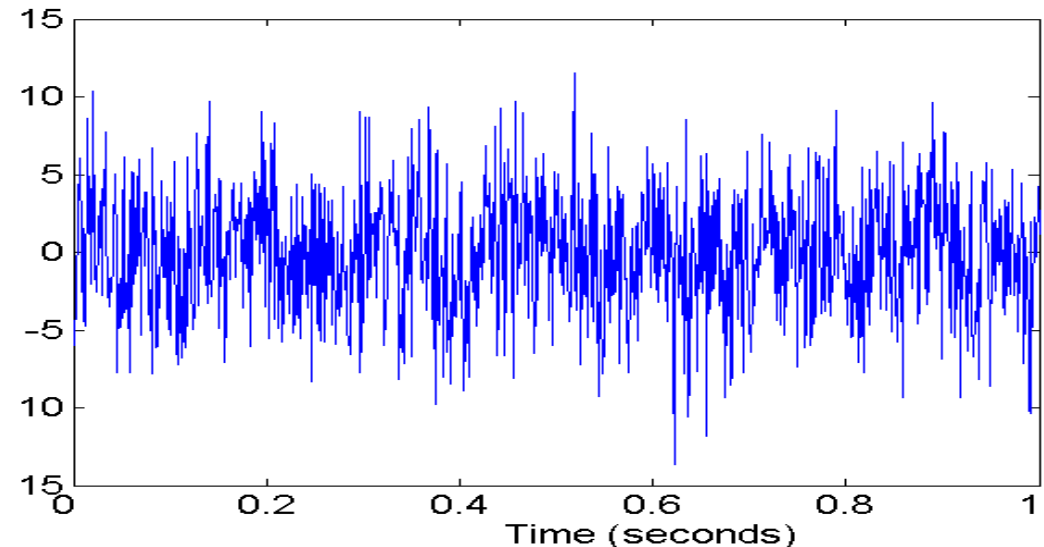
- duplicate records
- incomplete data
- inconsistent data

Noise

- Noise refers to modification of original values
 - Examples: distortion of a person's voice when talking on a poor phone and “snow” on television screen



Two Sine Waves



Two Sine Waves + Noise

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, etc.

Regression

- smooth by fitting the data into regression functions

Clustering

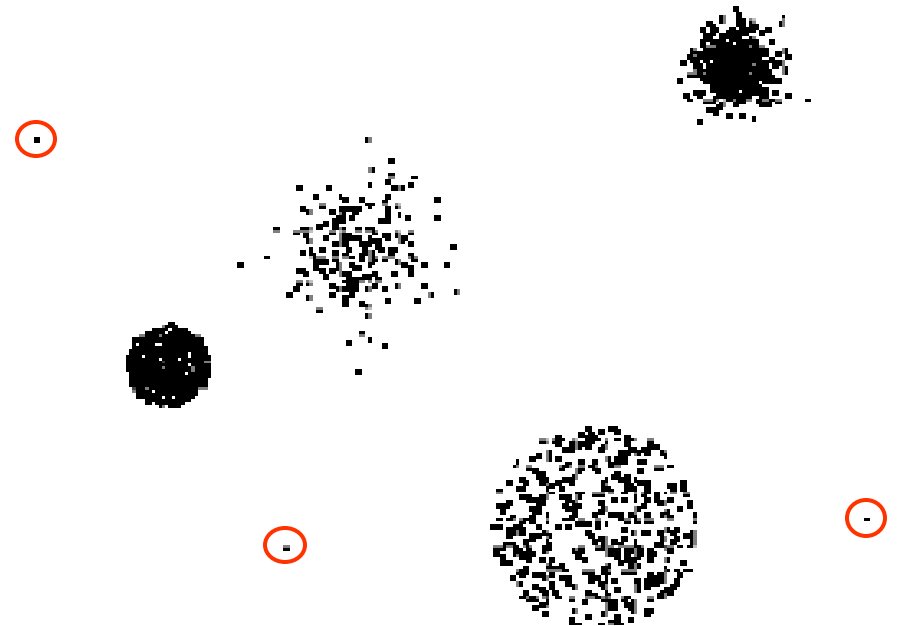
- detect and remove outliers

Combined computer and human inspection

- detect suspicious values and check by human (e.g., deal with possible outliers)

Outliers

- Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set



Distinguish between noise and outliers

1. Is noise ever interesting or desirable? Outliers?
2. Can noise objects be outliers?
3. Are noise objects always outliers?
4. Are outliers always noise objects?
5. Can noise make a typical value into an unusual one, or vice versa?

For each of the following data sets, explain whether or not data privacy is an important issue.

- Census data collected from 1900–1950.
- IP addresses and visit times of Web users who visit your Website.
- Images from Earth-orbiting satellites.
- Names and addresses of people from the telephone book.
- Names and email addresses collected from the Web.

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 relationship 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., Potter's Wheels)

DATA PREPROCESSING

In two aspects:

- 1. selecting data objects and attributes for the analysis*
- 2. for creating/changing the attributes.*

Aggregation

Sampling

Dimensionality
reduction

Feature subset
selection

Feature
creation

Discretization
and
binarization

Variable
transformation

Aggregation

- The combining of two or more objects into a single object.
- One way to aggregate transactions for this data set is to replace all the transactions of a single store with a single storewide transaction.
- This reduces the hundreds or thousands of transactions that occur daily at a specific store to a single daily transaction, and the number of data objects per day is reduced to the number of stores.

Table 2.4. Data set containing information about customer purchases.

Transaction ID	Item	Store Location	Date	Price	...
⋮	⋮	⋮	⋮	⋮	
101123	Watch	Chicago	09/06/04	\$25.99	...
101123	Battery	Chicago	09/06/04	\$5.99	...
101124	Shoes	Minneapolis	09/06/04	\$75.00	...
⋮	⋮	⋮	⋮	⋮	

Interpretation

The data can also be viewed as a multidimensional array,



Aggregation is the process of eliminating attributes,

Such as the type of item, or
reducing the number of values for a
particular attribute


e.g., reducing the possible values
for date from 365 days to 12
months.



This type of aggregation is commonly used in Online Analytical Processing (OLAP).

Aggregation

Combining two or more attributes (or objects) into a single attribute (or object)



Purpose

Data reduction

- Reduce the number of attributes or objects

Change of scale

- Cities aggregated into regions, states, countries, etc

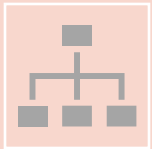
More “stable” data

- Aggregated data tends to have less variability

Advantages of aggregation



The smaller data sets resulting from data reduction require less memory and processing time



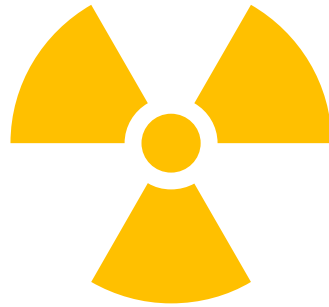
Aggregation can act as a change of scope or scale by providing a high-level view of the data instead of a low-level view.

In the previous example, aggregating over store locations and months gives us a monthly, per store view of the data instead of a daily, per item view.



the behaviour of groups of objects or attributes is often more stable than that of individual objects or attributes.

Disadvantage of aggregation



The potential loss of interesting details.



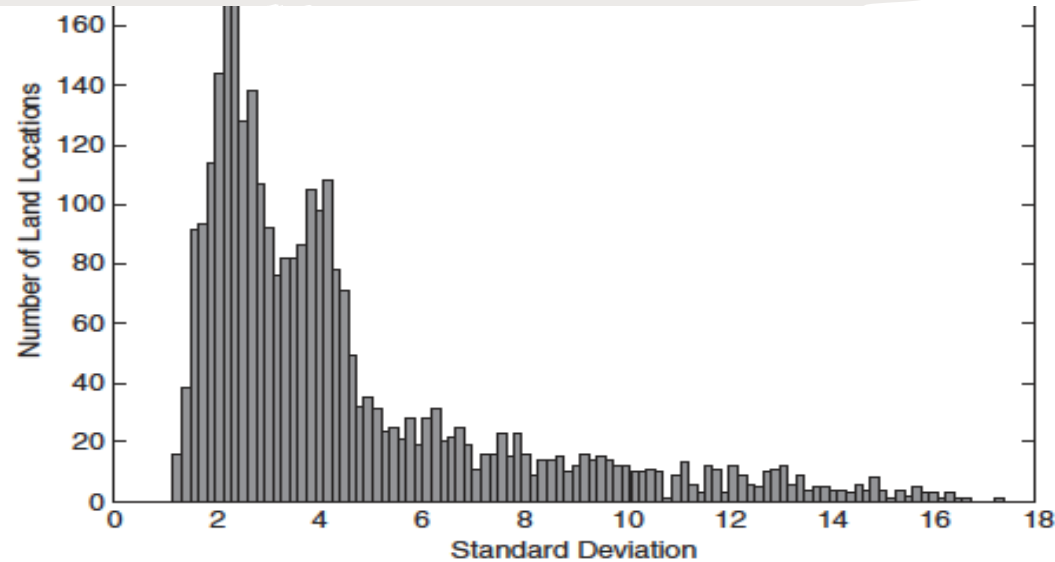
In the store example, aggregating over months loses information about which day of the week has the highest sales.

Example- Australian Precipitation

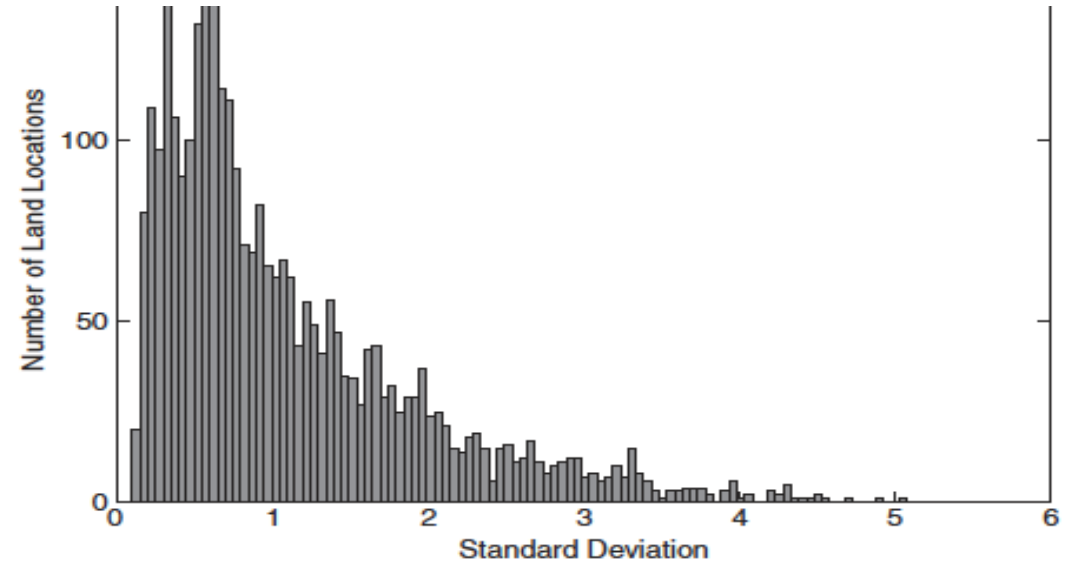
- This example is based on precipitation in Australia from the period 1982–1993.
- Figure 2.8(a) shows a histogram for the standard deviation of average monthly precipitation for 3,030 0.5° by 0.5° grid cells in Australia
- Figure 2.8(b) shows a histogram for the standard deviation of the average yearly precipitation for the same locations. The average yearly precipitation has less variability than the average monthly precipitation.
- All precipitation measurements (and their standard deviations) are in centimetres.



Example- Australian Precipitation



(a) Histogram of standard deviation of average monthly precipitation



(b) Histogram of standard deviation of average yearly precipitation

Figure 2.8. Histograms of standard deviation for monthly and yearly precipitation in Australia for the period 1982–1993.

Sampling



Sampling is the main technique employed for data selection.

It is often used for both the preliminary investigation of the data and the final data analysis.



Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.



Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.

Sampling ...

- The key principle for effective sampling is the following:
 - using a sample will work almost as well as using the entire data sets, if the sample is representative
 - A sample is representative if it has approximately the same property (of interest) as the original set of data

Types of Sampling



Sampling without replacement As each item is selected, it is removed from the population



Sampling with replacement

Objects are not removed from the population as they are selected for the sample.

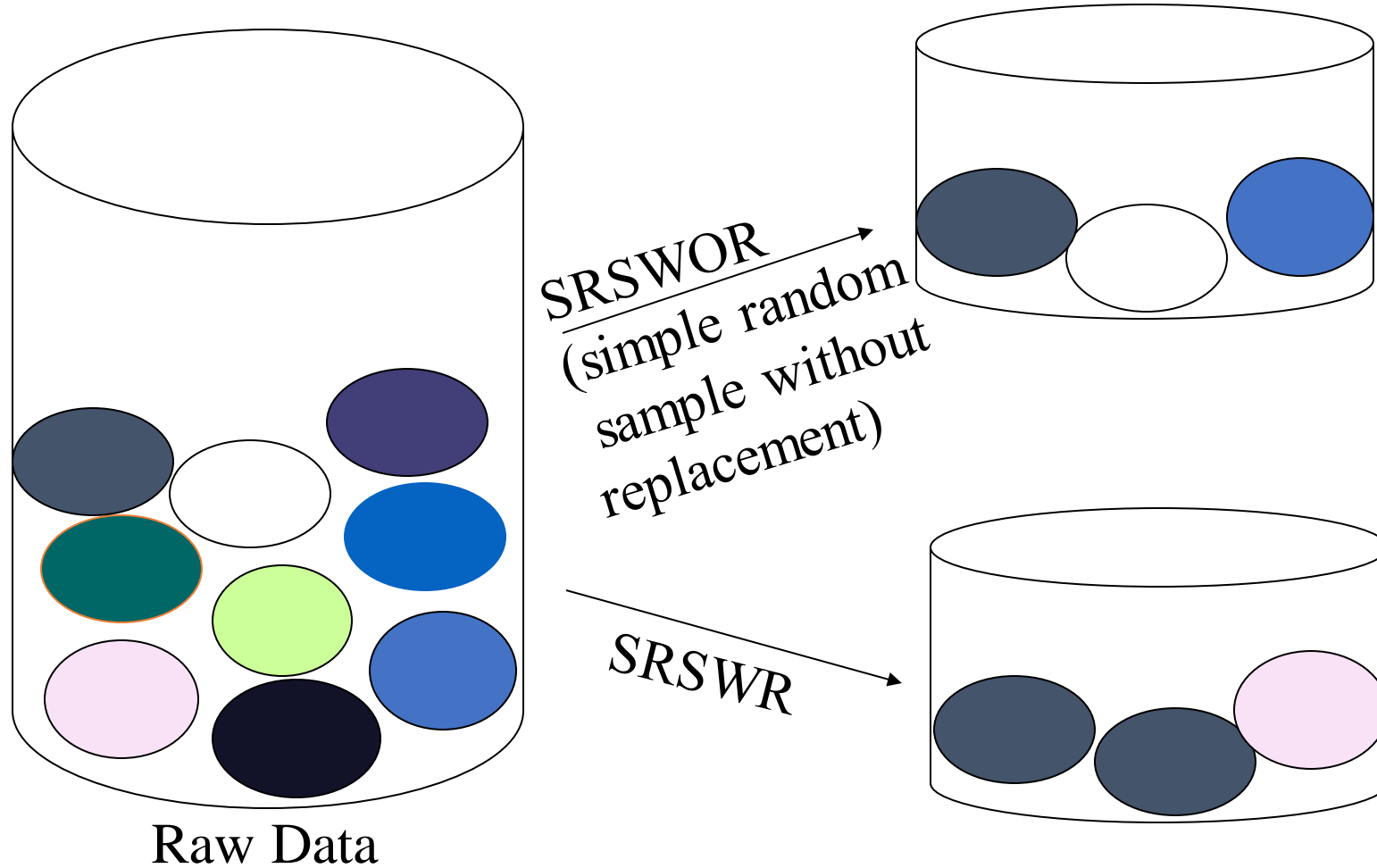
- In sampling with replacement, the same object can be picked up more than once



Stratified sampling

Split the data into several partitions; then draw random samples from each partition

Sampling: With or without Replacement



EXAMPLE - Sampling and Loss of Information

Once a sampling technique has been selected, it is still necessary to choose the sample size.

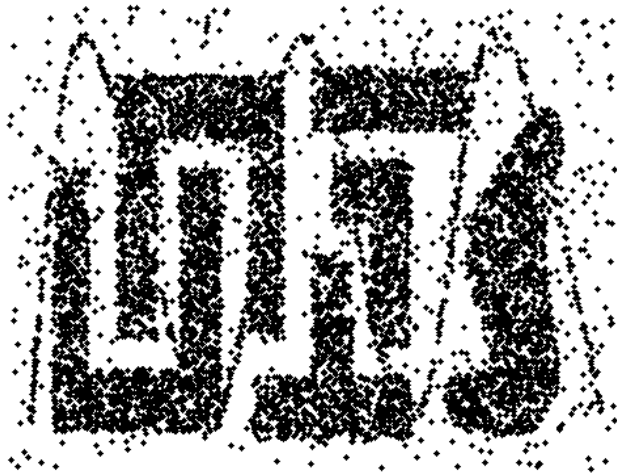
Larger sample sizes increase the probability that a sample will be representative, but they also eliminate much of the advantage of sampling.

Conversely, with smaller sample sizes, patterns can be missed, or erroneous patterns can be detected.

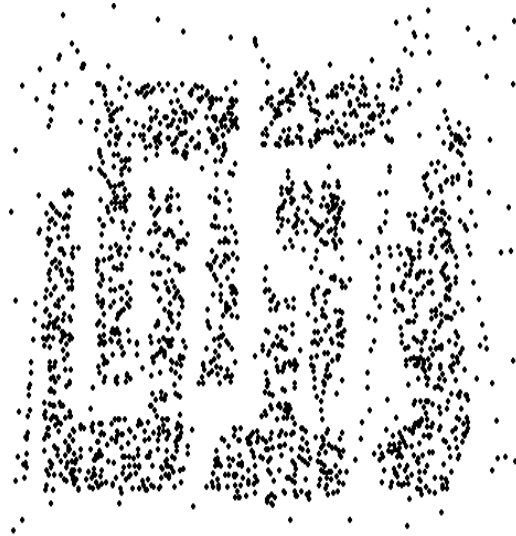
Figure 2.9(a) shows a data set that contains 8000 two-dimensional points, while Figures 2.9(b) and 2.9(c) show samples from this data set of size 2000 and 500, respectively.

Although most of the structure of this data set is present in the sample of 2000 points, much of the structure is missing in the sample of 500 points.

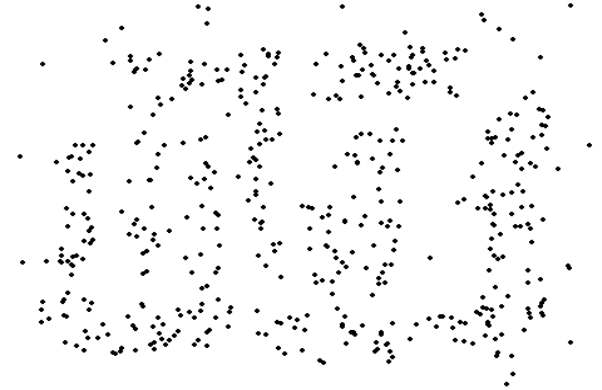
Sample Size



(a) 8000 points



(b) 2000 Points



(c) 500 Points

Example of the loss of structure with sampling.



DIMENSIONALITY REDUCTION

IMPORTANCE



Nature of Data Sets:

Data sets can often be characterized by the presence of a large number of features or attributes. These



Time Series Data Set Example:

Another example is a data set consisting of time series data, like daily closing prices of various stocks over a long period, say 30 years.



Challenges of High-Dimensional Data:

1. Curse of Dimensionality
2. Increased Complexity
3. Overfitting
4. Dimensionality Reduction



Importance of Feature Selection and Engineering:



Domain-Specific Considerations:

KEY BENEFITS



Many data mining algorithms work better if the dimensionality—the number of attributes in the data—is lower.



A reduction of dimensionality can lead to a more understandable model



It may allow the data to be more easily visualized.



The amount of time and memory required by the data mining algorithm is reduced with a reduction in dimensionality

Curse of Dimensionality

Challenges and Solutions in High-Dimensional Data Analysis

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies
- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful

Increased Sparsity

Implications for Classification

Implications for Clustering

The Role of Distance Metrics

Dimensionality Reduction Techniques

Feature Selection and Engineering

Specialized Algorithms

- support vector machines (SVMs)

Example: Recommender Systems

A recommender system for an e-commerce platform.

Your goal is to recommend products to users based on their past purchase history and browsing behaviour.

To make accurate recommendations, you consider various features about each product and user, including:

User Features:

- Age
- Gender
- Location
- Purchase history (number of items purchased)
- Average rating given to products

Product Features:

- Price
- Category
- Manufacturer
- Customer reviews (number of reviews, average rating)

Example: Recommender Systems (cont...)

However, as the e-commerce platform grows, the number of available products and users increases significantly.



Include even more features to improve recommendation accuracy:



Additional Features

User Features

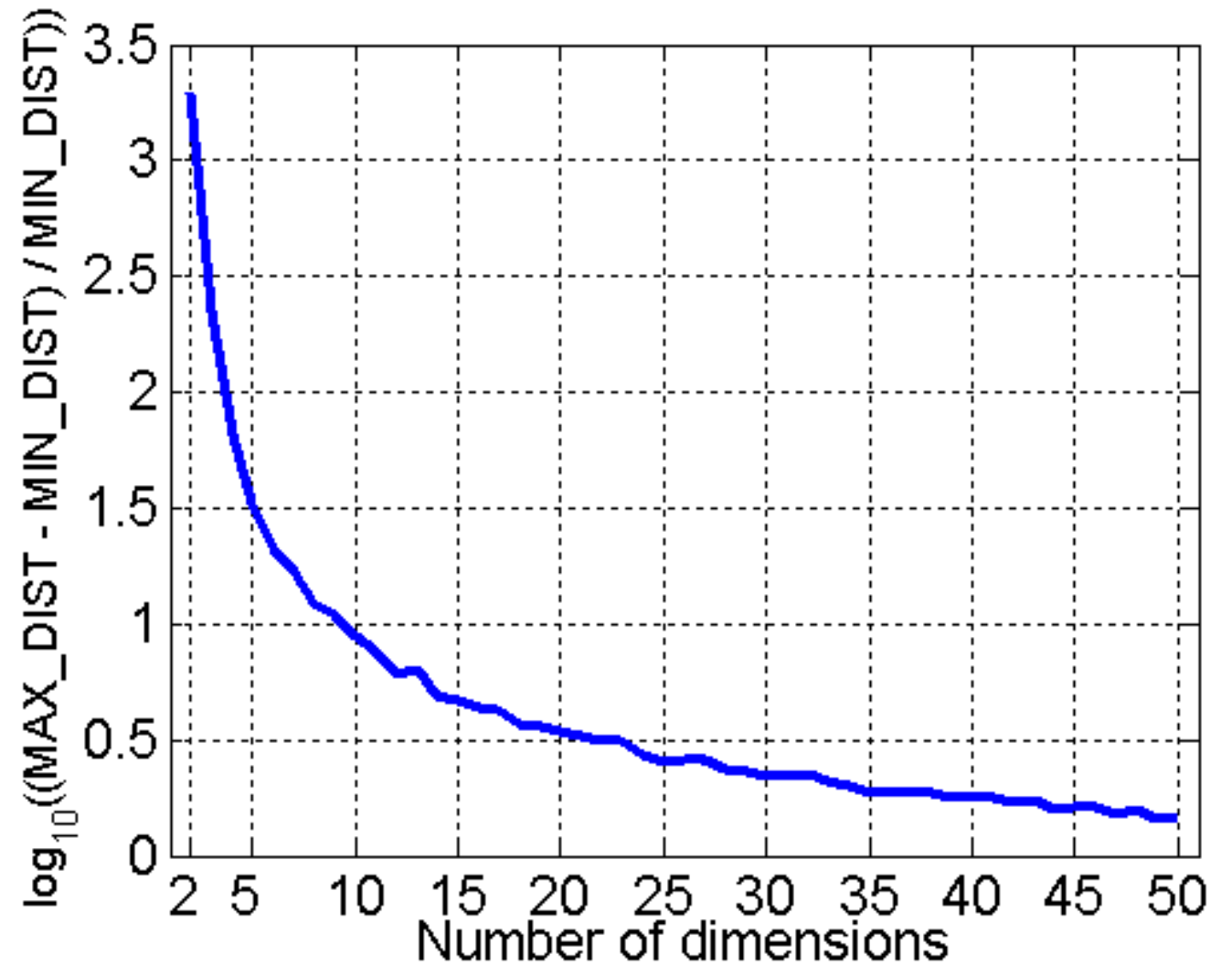
- Browsing history (pages visited, time spent on pages)
- Click-through rate on recommended products
- Social media activity related to the platform

Product Features:

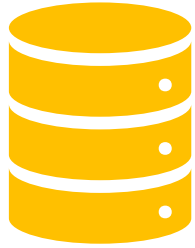
- Product image analysis (visual features)
- Text sentiment analysis of customer reviews
- Seasonal trends and discounts

Example

- Randomly generate 500 points
- Compute difference between max and min distance between any pair of points



Dimensionality Reduction



Purpose:

Avoid curse of dimensionality

Reduce amount of time and memory required by data mining algorithms

Allow data to be more easily visualized

May help to eliminate irrelevant features or reduce noise



Techniques

Principle Component Analysis

Singular Value Decomposition

Others: supervised and non-linear techniques

Dimensionality Reduction: PCA (*Principal Components Analysis*)



A linear algebra technique for continuous attributes that finds new attributes (principal components) that:

1. linear combinations of the original attributes
2. Orthogonal (perpendicular) to each other
3. They capture the maximum amount of variation in the data.



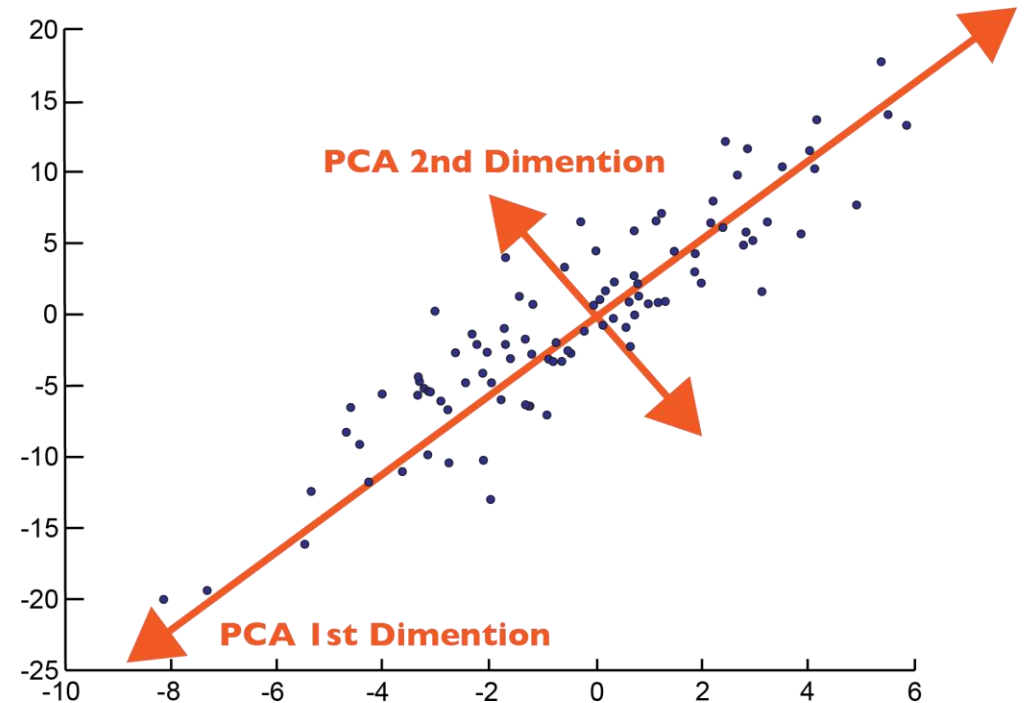
The first principal component captures the most variance, the second captures the second most, and so on.



This technique is particularly useful for continuous data.

Principle Component Analysis (PCA)

- n-dimensional data
- use first few orthogonal vectors (principal components)



Dimensionality Reduction: SVD (Singular Value Decomposition)



Singular Value Decomposition (SVD) is another powerful linear algebra technique closely related to PCA.



SVD decomposes a data matrix into three matrices, revealing the underlying structure of the data.



It is often used in dimensionality reduction and is valuable in scenarios where data might not be centered at the origin (unlike PCA, which assumes centered data).

Dimensionality Reduction: PCA (Principal Components Analysis)



Goal is to a projection that captures the largest amount of variation in find data



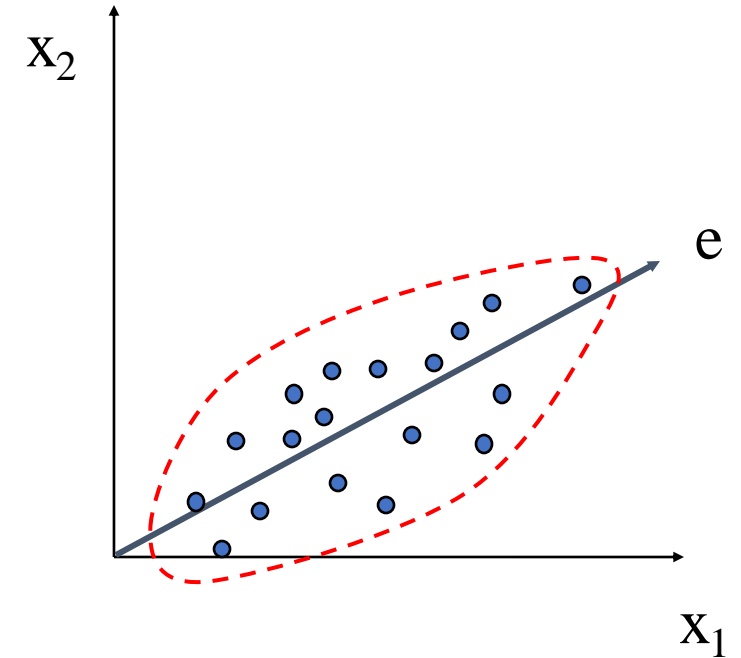
Find the m eigen vectors of the covariance matrix



The eigen vectors define the new space



Select only those m eigenvectors that contribute the most to the variation in the dataset ($m < n$)



Feature Subset Selection



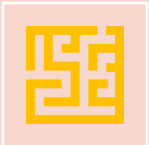
Another way to reduce dimensionality of data



Redundant features

duplicate much or all of the information contained in one or more other attributes

Example: purchase price of a product and the amount of sales tax paid



Irrelevant features

Contain no information that is useful for the data mining task at hand

Example: students' ID is often irrelevant to the task of predicting students' GPA

Feature Subset Selection Techniques:

Brute-force approach:

- Try all possible feature subsets as input to data mining algorithm

Embedded approaches:

- Feature selection occurs naturally as part of the data mining algorithm

Filter approaches:

- Features are selected before data mining algorithm is run

Wrapper approaches:

- Use the data mining algorithm as a black box to find best subset of attributes

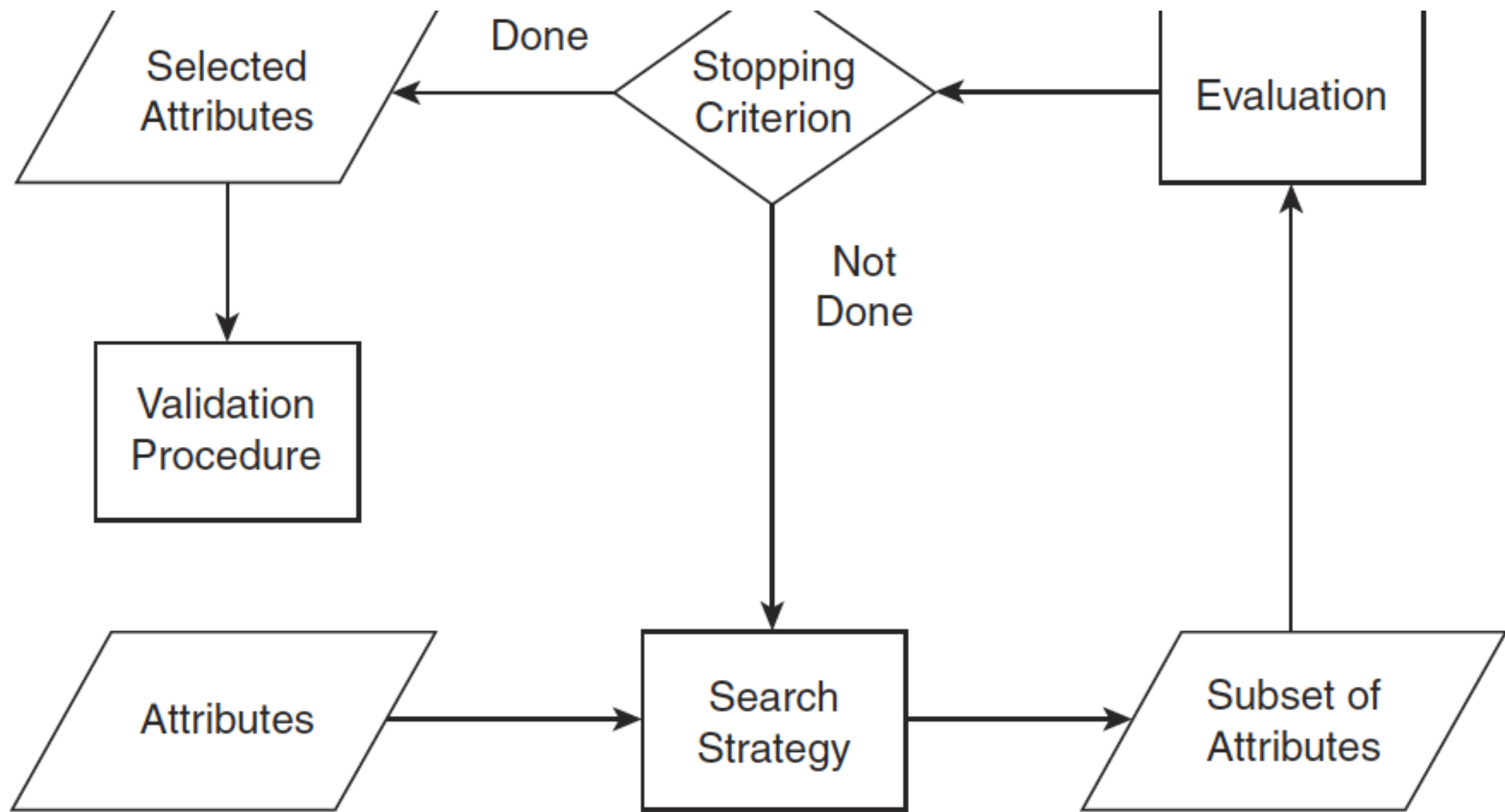


Figure 2.11. Flowchart of a feature subset selection process.

Feature Weighting

Feature weighting is an alternative to keeping or eliminating features.

More important features are assigned a higher weight, while less important features are given a lower weight.

These weights are sometimes assigned based on domain knowledge about the relative importance of features.

Alternatively, they can sometimes be determined automatically.

For example, some classification schemes, such as support vector machines produce classification models in which each feature is given a weight.

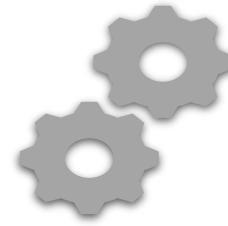
Features with larger weights play a more important role in the model.

The normalization of objects that takes place when computing the cosine similarity can also be regarded as a type of feature weighting.

Feature Creation



Create new attributes that can capture the important information in a data set much more efficiently than the original attributes



Three general methodologies:

1. Feature Extraction
 - domain-specific
2. Mapping Data to New Space
3. Feature Construction
 - combining features

Feature Extraction



The creation of a new set of features from the original raw data is known as feature extraction.



Consider a set of photographs, where each photograph is to be classified according to whether it contains a human face.



The raw data is a set of pixels, and as such, is not suitable for many types of classification algorithms.

EXAMPLE – DENSITY

- Consider a data set consisting of information about historical artifacts, which, along with other information, contains the volume and mass of each artifact.
- For simplicity, assume that these artifacts are made of a small number of materials (wood, clay, bronze, gold) and that we want to classify the artifacts with respect to the material of which they are made.
- In this case, a density feature constructed from the mass and volume features, i.e.,

$$\text{density} = \text{mass/volume}$$

would most directly yield an accurate classification.

Mapping the Data to a New Space



Fourier transform

there are a number of periodic patterns and a significant amount of noise, then these patterns are hard to detect



Wavelet transform :

for time series and other types of data.

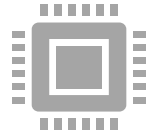
Mapping the Data to a New Space



Importance of a Different View

Sometimes, looking at data from a completely different angle can help us discover important patterns and features that might not be evident in the original representation.

Consider time series data, which frequently contains periodic patterns.



Time Series Data

Time series data often exhibits periodic behavior.

Detecting these patterns can be straightforward when there's only one clear periodic pattern with minimal noise.



Challenges with Multiple Patterns and Noise

However, things become complex when there are multiple periodic patterns overlaid with significant noise.

In such cases, detecting individual patterns can be challenging.



The Role of Fourier Transform

It allows us to transform time series data into a new representation where frequency information is explicit.

This transformation can make it easier to identify and analyse periodic patterns.

Example: Fourier Analysis

The Time Series

- In Figure 2.12(b), we have a time series that is the sum of three other time series.
- Two of these component time series, with frequencies of 7 and 17 cycles per second, are shown in Figure 2.12(a).
- The third time series represents random noise.

Power Spectrum

- After applying a Fourier transform to the original time series, we can compute a power spectrum.
- The power spectrum essentially quantifies the strength of different frequency components.
- In simpler terms, it tells us which frequencies are prominent in the data.

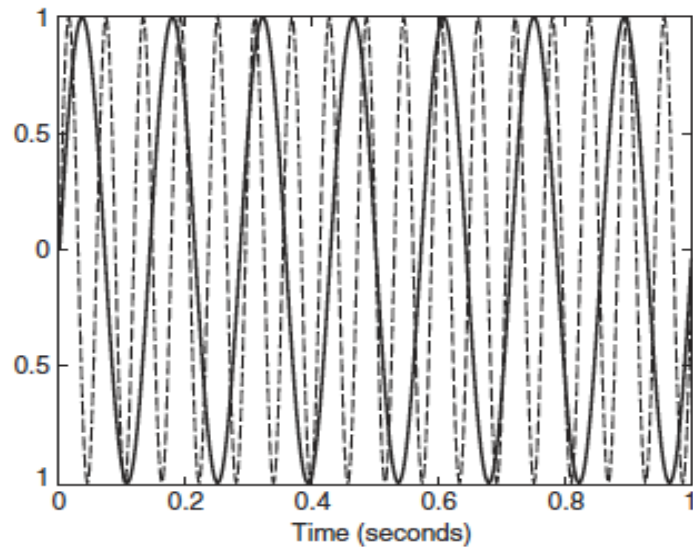
Identifying Patterns in the Power Spectrum

- Despite the presence of noise, the power spectrum often reveals clear peaks corresponding to the periodic patterns in the original time series.
- In our example, two peaks emerge, aligning with the frequencies of the two non-noisy component time series (7 and 17 cycles per second).

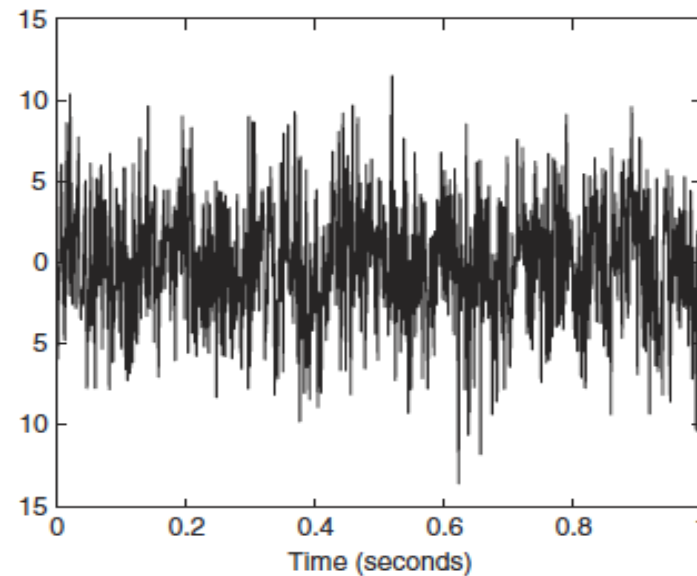
The Key Takeaway

- The main takeaway here is that by mapping the data to a new space, in this case, through a Fourier transform, we can obtain a set of attributes related to frequencies.
- These attributes can simplify the identification of important patterns and features in the data, even in the presence of noise.

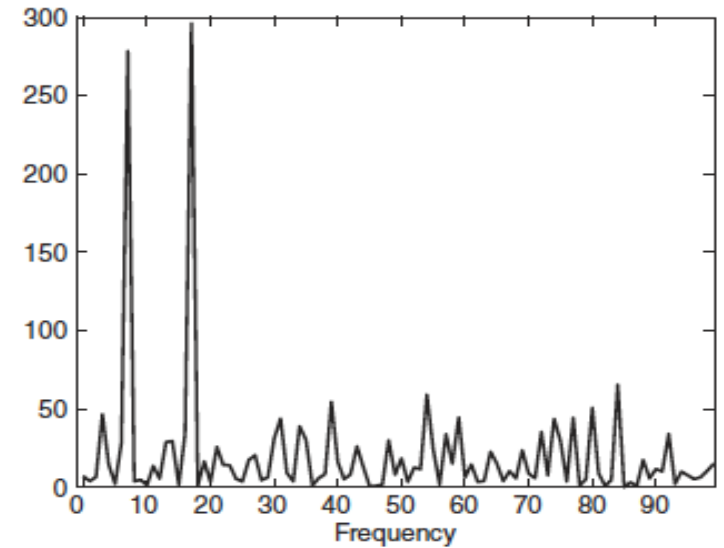
Mapping Data to a New Space



(a) Two time series.



(b) Noisy time series.



(c) Power spectrum.

Figure 2.12. Application of the Fourier transform to identify the underlying frequencies in time series data.

Discretization

Discretization is the process of converting continuous attributes into categorical ones.

It involves partitioning the range of a continuous attribute into intervals (bins) and assigning discrete labels to data points based on which interval they fall into.

When is Discretization Needed?

- When working with algorithms that require categorical attributes.
- For visualizing data or simplifying complex relationships.

Discretization



Three types of attributes

Nominal—values from an unordered set, e.g., color, profession

Ordinal—values from an ordered set, e.g., military or academic rank

Numeric—real numbers, e.g., integer or real numbers



Discretization: Divide the range of a continuous attribute into intervals

Interval labels can then be used to replace actual data values

Reduce data size by discretization

Supervised vs. unsupervised

Split (top-down) vs. merge (bottom-up)

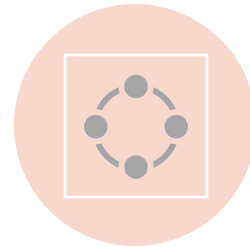
Discretization can be performed recursively on an attribute

Prepare for further analysis, e.g., classification

Data Discretization Methods



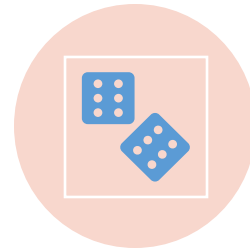
Binning (Top-down split, unsupervised)



Clustering analysis
(unsupervised, top-down split or bottom-up merge)



Decision-tree analysis
(supervised, top-down split)



Correlation (e.g., χ^2)
analysis (unsupervised, bottom-up merge)

Discretization of Continuous Attributes

*Discretization involves
two primary subtasks:*

1. Deciding the Number of Categories (n)

- The first step is to determine how many categories (intervals) we should have for our discretized attribute.
- This decision is crucial and impacts the granularity of the discretization.

2. Mapping Values to Categories

- Once we've decided on the number of categories, we need to determine how to map the values of the continuous attribute to these categories.
- This step involves dividing the range of values into intervals and assigning a categorical label to each interval.

Split Points in Discretization



The key challenge in discretization is deciding where to place the split points that divide the continuous attribute into intervals.



Split points are thresholds that separate one interval from another.



After sorting the continuous attribute values, we place $n-1$ split points to create n intervals.

Representing Discretization Results

Intervals Representation:

- This representation defines the intervals explicitly.
- For example: $\{(x_0, x_1], (x_1, x_2], \dots, (x_{n-1}, x_n)\}$, where x_0 and x_n can be $\pm\infty$.

Inequalities Representation:

- Alternatively, you can represent the discretization as a series of inequalities, e.g., $x_0 < x \leq x_1, \dots, x_{n-1} < x < x_n$.

Unsupervised Discretization

Equal-width (distance) partitioning

- Divides the range into N intervals of equal size: uniform grid
- if A and B are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
- The most straightforward, but outliers may dominate presentation
- Skewed data is not handled well

Equal-depth (frequency) partitioning

- Divides the range into N intervals, each containing approximately same number of samples
- Good data scaling
- Managing categorical attributes can be tricky

Clustering-Based Discretization

- Clustering methods, such as K-means clustering, can be applied to discretize continuous attributes.
- K-means identifies cluster centroids and assigns data points to the nearest centroid, effectively grouping them.
- The cluster boundaries can be used as split points for discretization.

Visual Inspection

- Sometimes, the most effective approach is to visually inspect the data.
- By plotting the data, patterns and natural boundaries can be identified, allowing for manual discretization decisions.

Discretization
Without Using
Class Labels
(Binning vs.
Clustering)

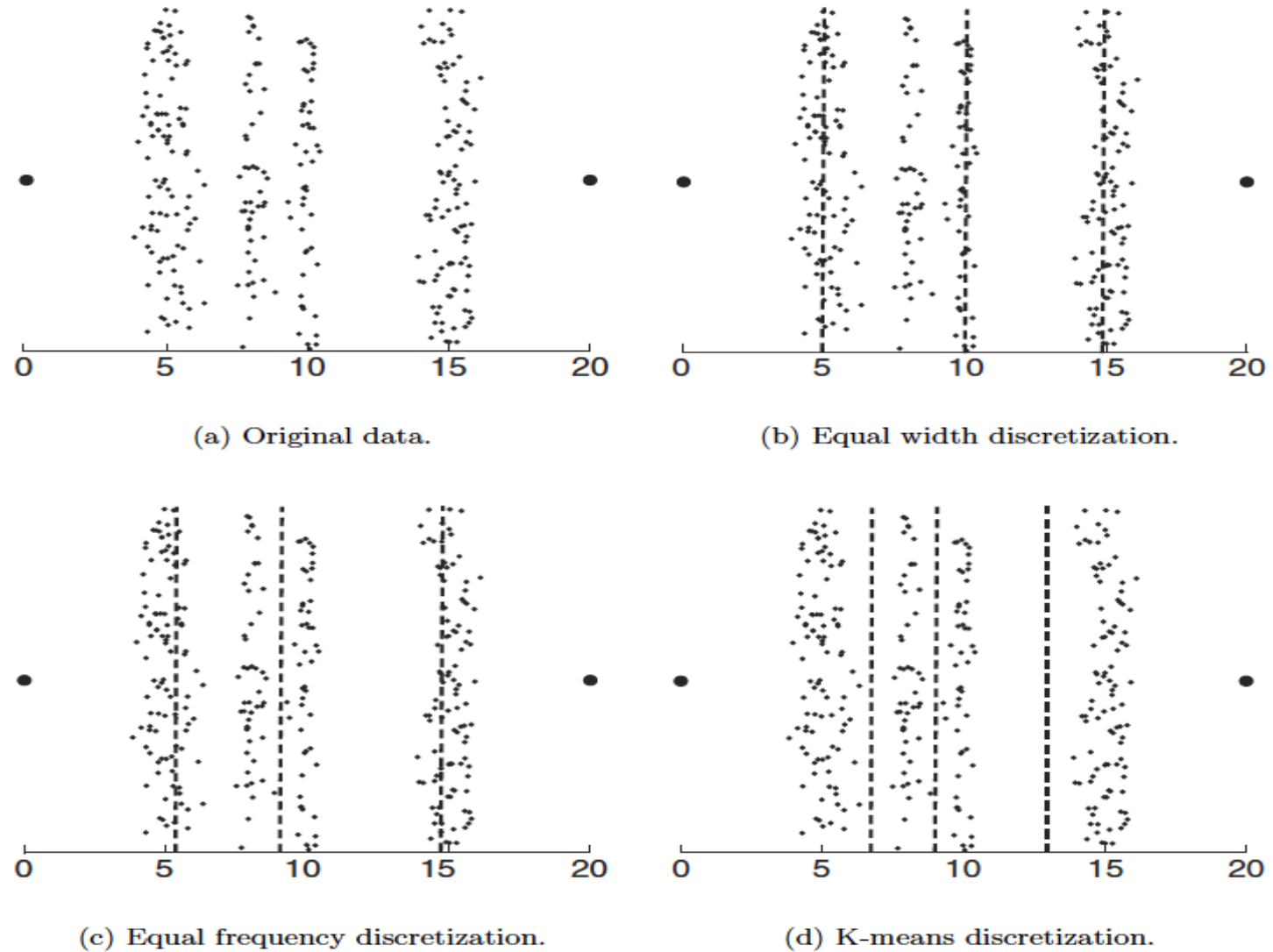


Figure 2.13. Different discretization techniques.

Supervised Discretization

The Importance of Class Labels

- In classification tasks, having knowledge of class labels can enhance discretization.
- An interval constructed without class labels may contain a mixture of different class labels.
- We aim to create intervals that maximize the purity, meaning each interval primarily contains a single class label.

Conceptual Approach: Maximizing Purity

- One simple approach is to place splits in a way that maximizes the purity of the intervals.
- Purity refers to the extent to which an interval contains predominantly one class label.
- However, this approach involves potentially arbitrary decisions about interval purity and minimum size.

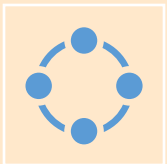
Statistically-Based Approaches



Statistically-based approaches offer more robust solutions.



They often begin with each attribute value in a separate interval and merge adjacent intervals that are statistically similar.



Two common approaches are:

Bottom-up: Start with individual values and merge similar intervals.

Top-down: Begin with all values in one interval and bisect it to minimize entropy. Repeat until a stopping criterion is met.

Entropy-Based Discretization

- Entropy measures the purity of an interval. Lower entropy indicates higher purity.
- For an interval with k different class labels and class probabilities, the entropy is calculated using the formula: +
- Let k be the number of different class labels, m_i be the number of values in the i^{th} interval of a partition, and m_{ij} be the number of values of class j in interval i .
- Then the entropy e_i of the i^{th} interval is given by the equation: •

$$e_i = - \sum_{j=1}^k p_{ij} \log_2 p_{ij},$$

- where $p_{ij} = m_{ij}/m_i$ is the probability (fraction of values) of class j in the i^{th} interval.
- The total entropy of a partition is a weighted average of individual interval entropies:
-

Entropy-Based Discretization

- The total entropy of a partition is a weighted average of individual interval entropies:

$$e = \sum_{i=1}^n w_i e_i,$$

- where m is the number of values, $w_i = m_i/m$ is the fraction of values in the i^{th} interval, and n is the number of intervals.

Example: Discretization of Two Attributes

We'll illustrate the top-down entropy-based method using a two-dimensional dataset.

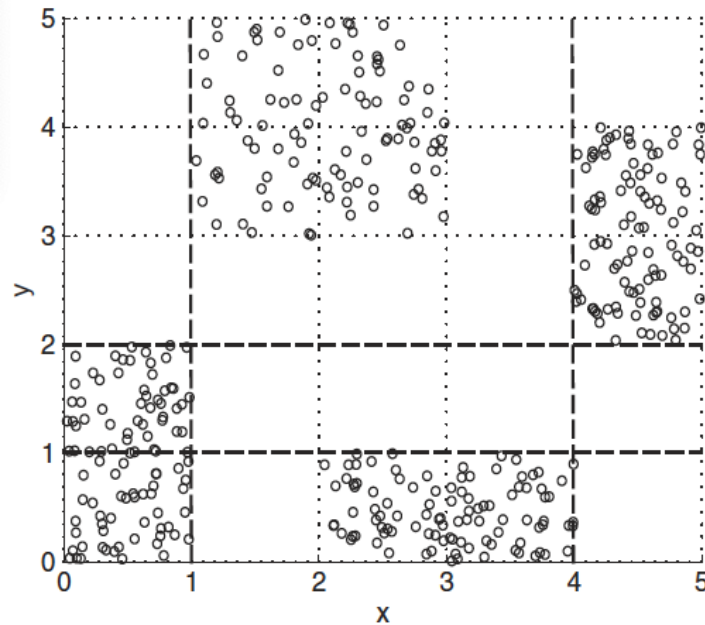
- The x and y attributes will be independently discretized into intervals.

The number of intervals and the quality of discretization will be :

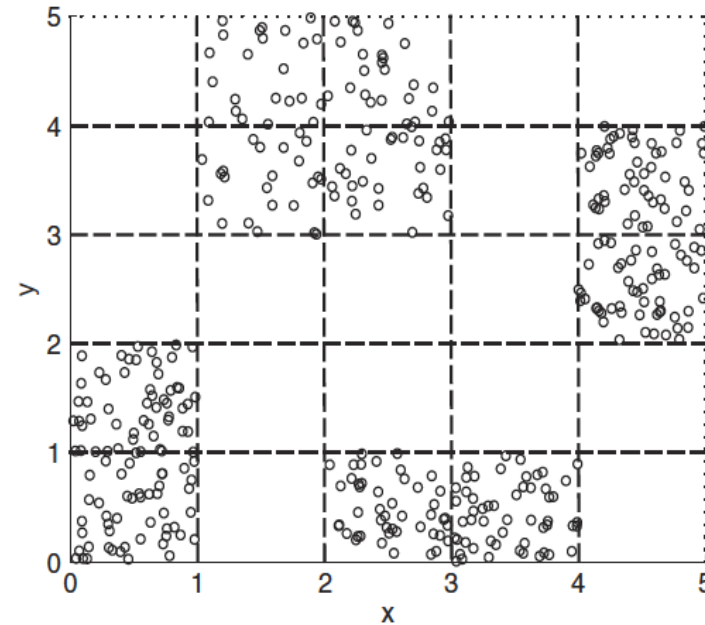
- In two dimensions, the classes of points are well separated, but in one dimension, this is not so.
- five intervals work better than three, but six intervals do not improve the discretization much, at least in terms of entropy.

Consequently, it is desirable to have a stopping criterion that automatically finds the right number of partitions.

Example: Discretization of Two Attributes



(a) Three intervals



(b) Five intervals

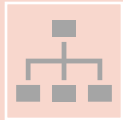
Figure 2.14. Discretizing x and y attributes for four groups (classes) of points.

Data Integration



Data integration:

Combines data from multiple sources into a coherent store



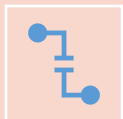
Schema integration: e.g.,
 $A.cust-id \equiv B.cust-\#$

Integrate metadata from different sources



Entity identification problem:

Identify real world entities from multiple data sources, e.g.,
Bill Clinton = William Clinton



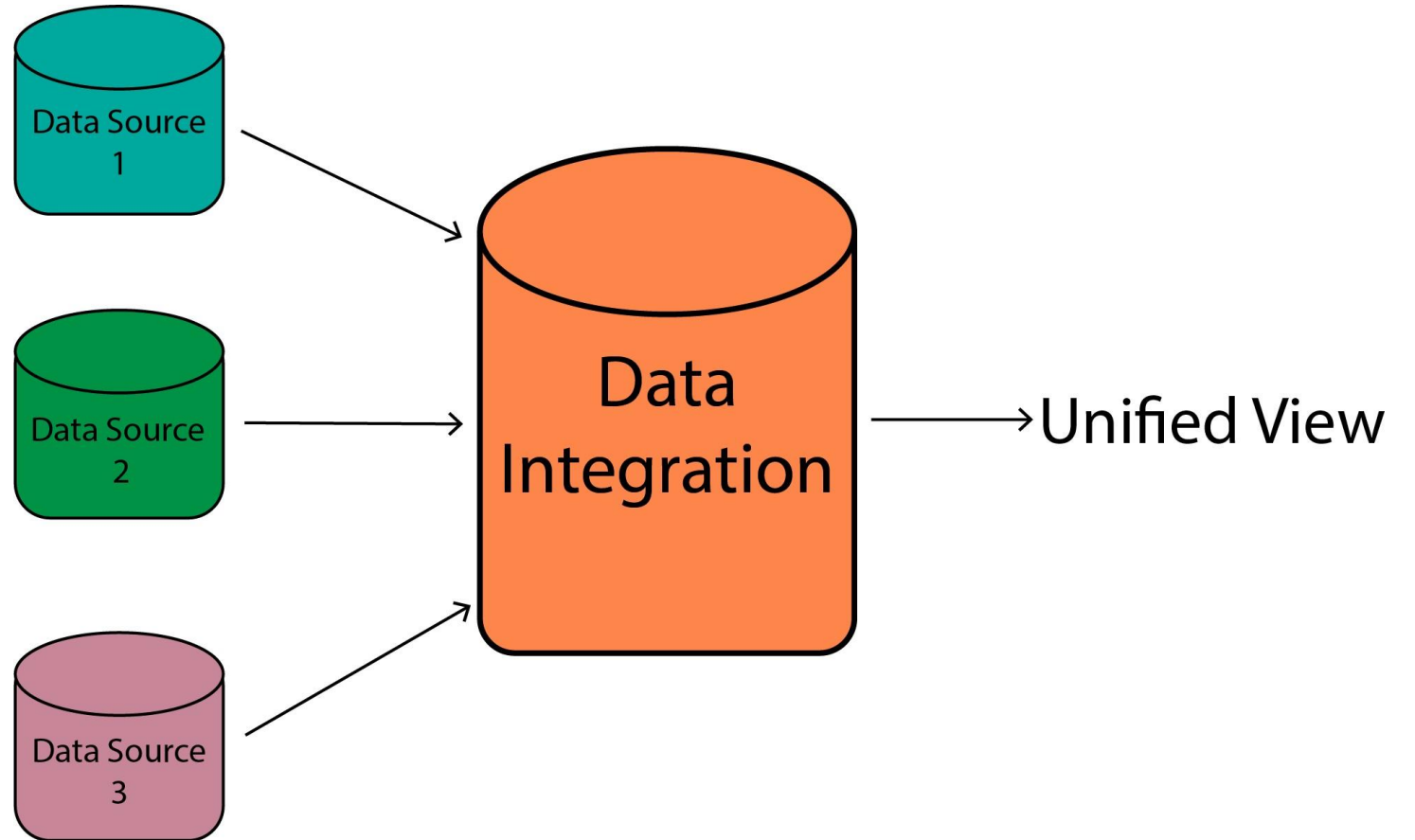
Detecting and resolving data value conflicts

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

Possible reasons: different representations, different scales, e.g., metric vs. British units

Data Integration

- Combines data from multiple sources
- Entity identification
- E.g., users, items
- Redundant data
- E.g., correlation analysis



Handling Redundancy in Data Integration

Redundant data occur often when integration of multiple databases

- *Object 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 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

Variable transformation



A transformation that is applied to all the values of a variable.



For example, if only the magnitude of a variable is important, then the values of the variable can be transformed by taking the absolute value.



It involves applying mathematical functions or normalization techniques to the values of a variable to change its distribution or scale.



Following types

Simple Functional Transformations:
Normalization or Standardization

Simple Functions



Simple mathematical functions are applied individually to each value of a variable. Examples of such transformations include:

x^k (raising to a power)

$\log(x)$ (logarithm)

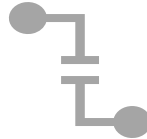
e^x (exponential)

\sqrt{x} (square root)

$1/x$ (reciprocal)

$\sin(x)$ (sine function)

$|x|$ (absolute value)



These transformations are often used to convert data with non-Gaussian (non-normal) distributions into data that approximate a normal distribution.



For example, logarithmic transformations are commonly used for this purpose.

Example

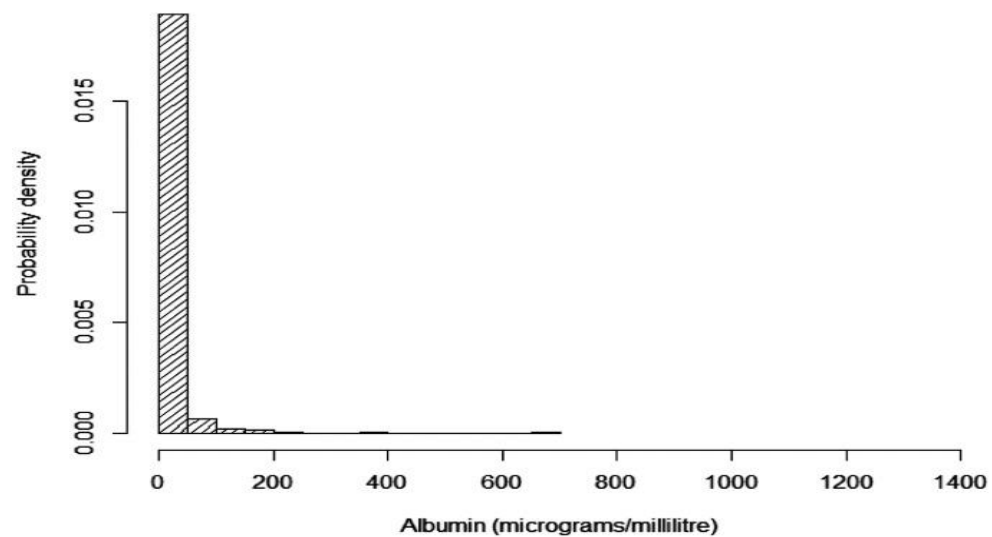
The advantage of common logarithms is that they are more readily 'interpreted' or checked.

For example, a \log_{10} value of '2. xxx' will lie between 100 and 1000 since $\log_{10}(100) = 2$ and $\log_{10}(1000) = 3$.

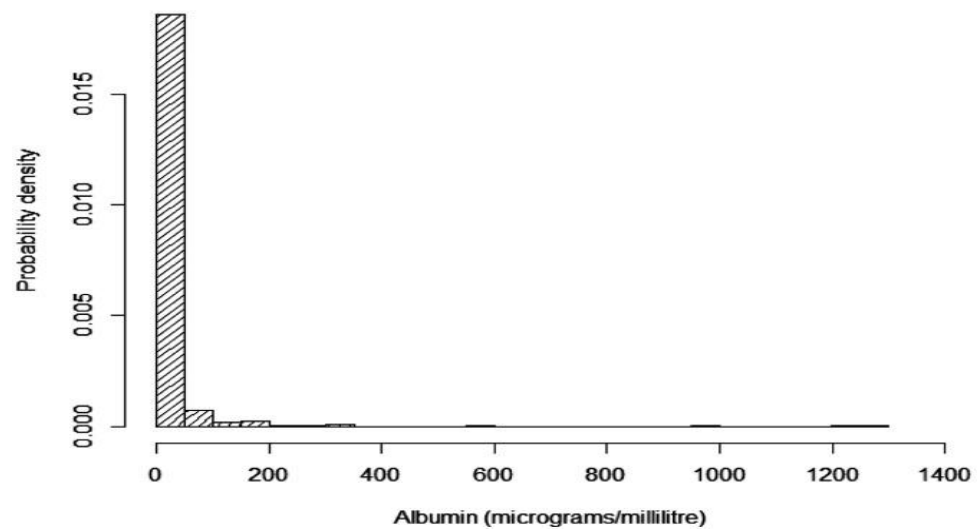
The transformed distributions, using a \log_{10} transformation, are shown in following figure.

This includes a fitted curve representing the normal distribution, with the same mean and standard deviation

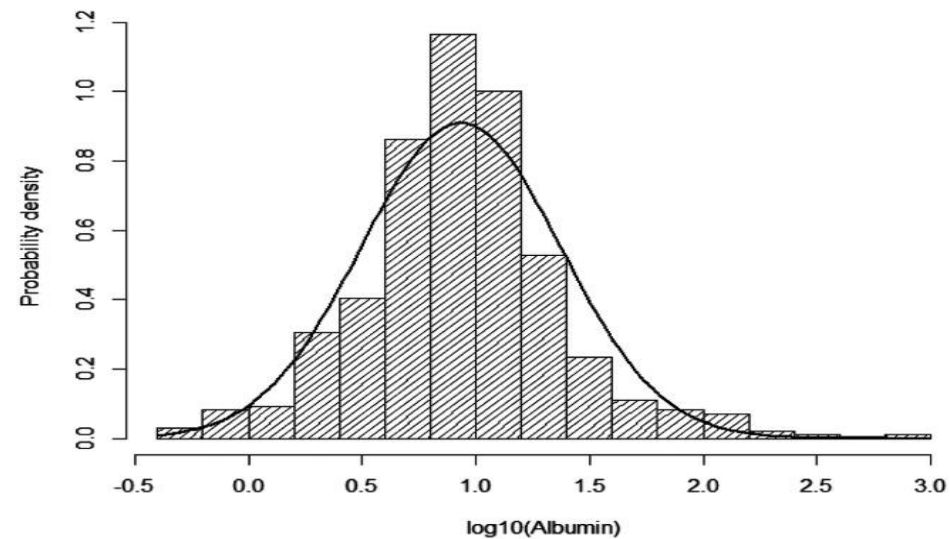
Urine albumin for males



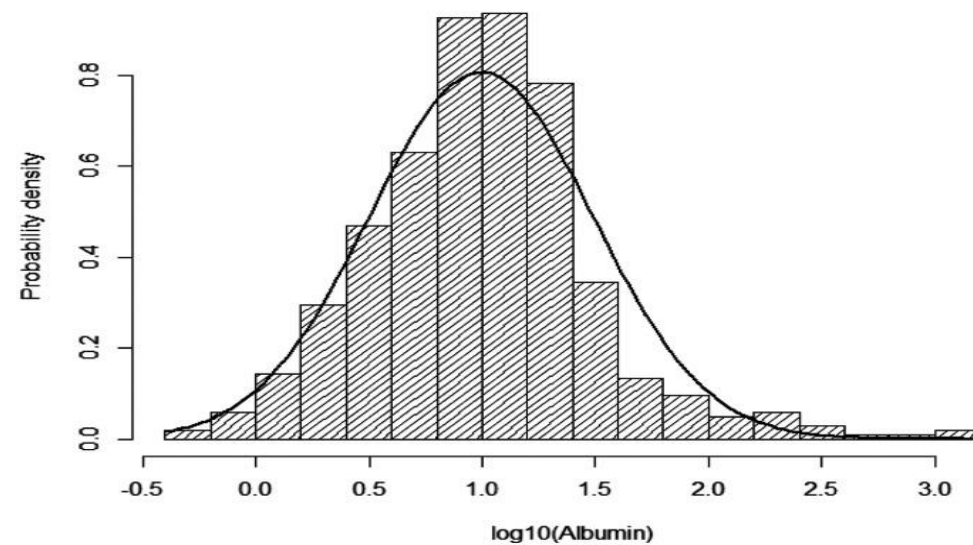
Urine albumin for females



Log10(albumin) for males



Log10(albumin) for females



Reasons for Variable Transformation:



Changing the scale of the data (e.g., using a log transformation to compress a wide range of values).



Making data more suitable for specific algorithms or analyses.



Handling outliers or extreme values.

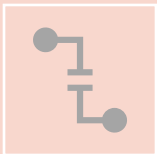


Achieving a desired property of the transformed attribute (e.g., maintaining order or ensuring all values are positive).

Caution in Applying Transformations:



Variable transformations should be applied with caution because they can change the nature of the data.



For instance, the transformation $1/x$ reduces the magnitude of values that are 1 or larger but increases the magnitude of values between 0 and 1.

Normalization or Standardization

- The goal of normalization or standardization is to make a set of values share a specific property
- Standardization typically involves subtracting the mean and dividing by the standard deviation,
 - resulting in a new variable with a mean of 0 and a standard deviation of 1.
- This is particularly useful when working with multiple variables to prevent variables with large values from dominating the analysis.
- If \bar{x} is the mean (average) of the attribute values and s_x is their standard deviation, then the transformation

$$x' = (x - \bar{x}) / s_x$$

- creates a new variable that has a mean of 0 and a standard deviation of 1.

Normalization

- **Min-max normalization:** to $[new_min_A, new_max_A]$

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

- Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0].

- Then \$73,000 is mapped to $\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0) + 0 = 0.716$

- **Z-score normalization** (μ : mean, σ : standard deviation):

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

- Ex. Let $\mu = 54,000$, $\sigma = 16,000$. Then $\frac{73,600 - 54,000}{16,000} = 1.225$

- **Normalization by decimal scaling**

$$v' = \frac{v}{10^j} \quad \text{Where } j \text{ is the smallest integer such that } \text{Max}(|v'|) < 1$$

Handling Outliers in Standardization:



The standardization method mentioned (using mean and standard deviation) can be sensitive to outliers.



In such cases, more robust measures like the median and absolute standard deviation can be used instead.



Modified transformation :

the mean is replaced by the median, i.e., the middle value.
Second, the standard deviation is replaced by the absolute standard deviation

Trending YouTube Video Statistics

- The Trending YouTube Video Statistics is a daily record with daily statistics for trending Youtube videos which were collected using YouTube API.
- It includes several months (and counting) of data on daily trending YouTube videos, with up to 200 listed trending videos per day.
- Each region's data is in a separate file.
- Data includes the video title, channel title, publish time, tags, views, likes and dislikes, description, and comment count.
- <https://www.kaggle.com/datasets/datasnaek/youtube-new>