

# Data Preprocessing

Jun Du

The University of Western Ontario

[jdu43@uwo.ca](mailto:jdu43@uwo.ca)

# Outline

- Data
- Data Preprocessing: An Overview
- Data Cleaning
- Data Transformation and Data Discretization
- Data Reduction
- Summary

# What is Data?

- Collection of **data objects** and their **attributes**
- Data objects → rows
- Attributes → columns

Objects

Attributes

<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

# Data Objects

- A **data object** represents an entity.
- Examples:
  - Sales database: customers, store items, sales
  - Medical database: patients, treatments
  - University database: students, professors, courses
- Also called *examples, instances, records, cases, samples, data points, objects, etc.*
- Data objects are described by **attributes**.

# Attributes

- An **attribute** is a data field, representing a characteristic or feature of a data object.
- Example:
  - Customer Data: customer\_ID, name, gender, age, address, phone number, etc.
  - Product data: product\_ID, price, quantity, manufacturer, etc.
- Also called *features, variables, fields, dimensions, etc.*

# Attribute Types (1)

- Nominal (Discrete) Attribute
  - Has only **a finite set of values** (such as, categories, states, etc.)
  - E.g., Hair\_color = {*black, blond, brown, grey, red, white, ...*}
  - E.g., marital status, zip codes
- Numeric (Continuous) Attribute
  - Has real numbers as attribute values
  - E.g., temperature, height, or weight.
- Question: what about student id, SIN, year of birth?

# Attribute Types (2)

- Binary
  - A special case of nominal attribute: **with only 2 states** (0 and 1)
  - Gender = {*male, female*};
  - Medical test = {*positive, negative*}
- Ordinal
  - Usually a special case of nominal attribute: values have a **meaningful order (ranking)**
  - Size = {*small, medium, large*}
  - Army rankings

# Outline

- Data
- **Data Preprocessing: An Overview**
- Data Cleaning
- Data Transformation and Data Discretization
- Data Reduction
- Summary



# Data Preprocessing

- Why preprocess the data?
  - Data quality is poor in real world.
  - No quality data, no quality mining results!
- Measures for data quality
  - Accuracy: noise, outliers, ...
  - Completeness: missing values, ...
  - Redundancy: duplicated data, irrelevant data, ...
  - Consistency: some modified but some not, ...
  - .....

# Typical Tasks in Data Preprocessing

- Data Cleaning
  - Handle missing values, noisy / outlier data, resolve inconsistencies, ...
- Data Transformation
  - Aggregation
  - Type Conversion
  - Normalization
- Data Reduction
  - Data Sampling
  - Dimensionality Reduction
- .....

# Outline

- Data
- Data Preprocessing: An Overview
- **Data Cleaning**
- Data Transformation and Data Discretization
- Data Reduction
- Summary

# Data Cleaning

- **Missing value**: lacking attribute values
  - E.g., *Occupation* = “ ”
- **Noise (Error)**: modification of original values
  - E.g., *Salary* = “-10”
- **Outlier**: considerably different from most of the other data (not necessarily error)
  - E.g., *Salary* = “2,100,000”
- **Inconsistency**: discrepancies in codes or names
  - E.g., *Age*=“42”, *Birthday*=“03/07/2010”
  - Was rating “1, 2, 3”, now rating “A, B, C”
- .....

# Missing Values

- Reasons for missing values
  - Information is not collected
    - E.g., people decline to give their age and weight
  - Attributes may not be applicable to all cases
    - E.g., annual income is not applicable to children
  - Human / Hardware / Software problems
    - E.g., Birthdate information is accidentally deleted for all people born in 1988.
  - .....

# How to Handle Missing Value?

- Eliminate \ ignore missing value
  - Eliminate \ ignore the examples
  - Eliminate \ ignore the features
  - Simple; not applicable when data is scarce
- Estimate missing value
  - Global constant : e.g., “unknown”,
  - Attribute mean (median, mode)
  - Predict the value based on features (data imputation)
    - Estimate gender based on first name (name gender)
    - Estimate age based on first name (name popularity)
    - Build a predictive model based on other features
  - Missing value estimation depends on the missing reason!

# Demonstration

- ReplaceMissingValues
  - \Weka\Vote
  - Replacing missing values for nominal and numeric attributes
- More functions in Rapidminer

# Noisy (Outlier) Data

- Noise: refers to modification of original values
- 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 (Outlier) Data?

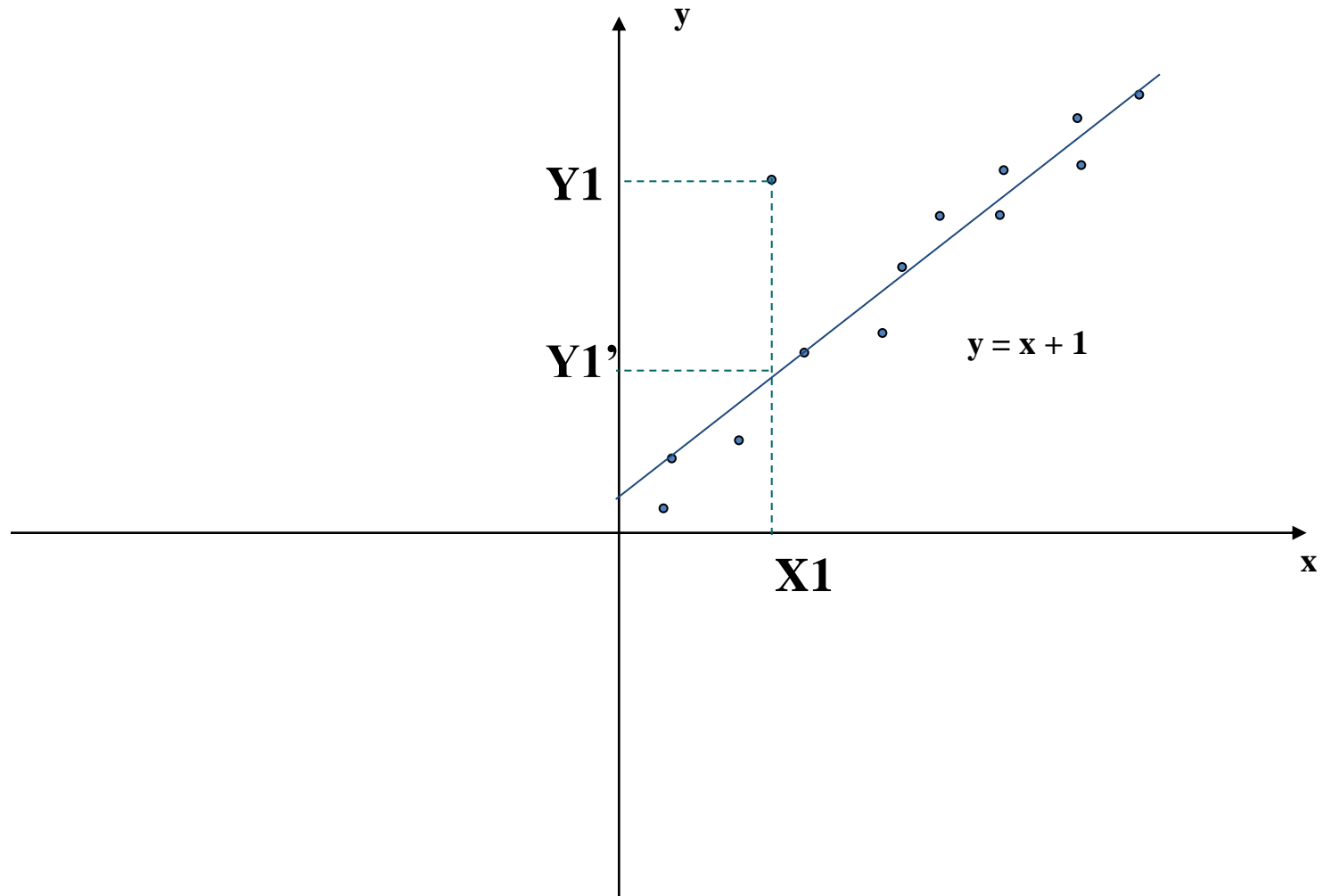
- Binning
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
  - smooth by fitting the data into regression functions
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human

# Binning

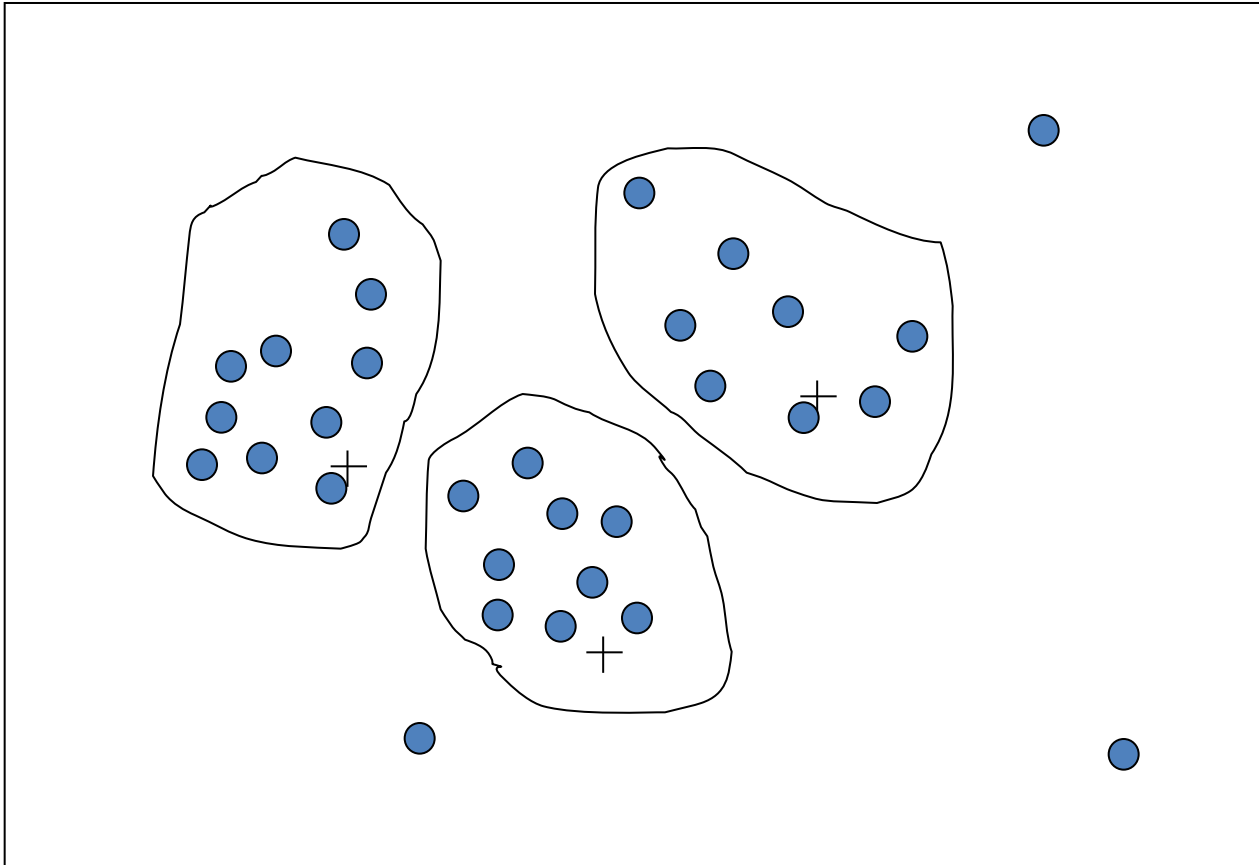
Sort data in ascending order: 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

- Partition into equal-frequency (equal-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

# Regression



# Cluster Analysis



# Outline

- Data
- Data Preprocessing: An Overview
- Data Cleaning
- Data Transformation and Data Discretization
- Data Reduction
- Summary

# Data Transformation

- **Aggregation:**
  - Attribute / example summarization
- **Feature type conversion:**
  - Nominal  $\leftrightarrow$  Numeric, ...
- **Normalization:**
  - Scaled to fall within a small, specified range
- **Attribute/feature construction:**
  - New attributes constructed from the given ones

# Aggregation

- Combining two or more attributes (examples) into a single attribute (example)
- Combining two or more attribute values into a single attribute value
- Purpose
  - Change of scale
    - Cities aggregated into regions, states, countries, etc
  - More “stable” data
    - Aggregated data tends to have less variability
  - More “predictive” data
    - Aggregated data might have high Predictability

# Demonstration

- MergeTwoValues
  - \Weka\contact-lenses
  - Merge class values “soft” and “hard”
- Effective aggregation in real-world application



# Feature Type Conversion

- Some algorithms can only handle numeric features; some can only handle nominal features. Only few can handle both.
- Features have to be converted to satisfy the requirement of learning algorithms.
  - Numeric  $\rightarrow$  Nominal (Discretization)
    - E.g., Age Discretization: Young 18-29; Career 30-40; Mid-Life 41-55; Empty-Nester 56-69; Senior 70+
  - Nominal  $\rightarrow$  Numeric
    - Introduce multiple numeric features for one nominal feature
    - Nominal  $\rightarrow$  Binary (Numeric)
    - E.g., size={L, M, S}  $\rightarrow$  size\_L: 0, 1; size\_M: 0, 1; size\_S: 0, 1

# Demonstration

- Discretize
  - \Weka\diabetes
  - Discretize “age” (equal bins vs equal frequency)
- NumericToNominal
  - \Weka\diabetes
  - Discretize “age” (vs “Discretize” method)
- NominalToBinary
  - \UCI\autos
  - Convert “num-of-doors”
  - Convert “drive-wheels”

# Normalization

Scale the attribute values to a small specified range

- Min-max normalization: to  $[new\_min_A, new\_max_A]$

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

- E.g., 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 ( $\mu$ : mean,  $\sigma$ : standard deviation):
- .....

# Demonstration

- Normalize
  - \Weka\diabetes
  - Normalize “age”
- Standardize
  - \Weka\diabetes
  - Standardize “age” (vs “Normalize” method)

# Outline

- Data
- Data Preprocessing: An Overview
- Data Cleaning
- Data Transformation and Data Discretization
- **Data Reduction**
- Summary

# Sampling

- Big data era: **too expensive (or even infeasible)** to process the entire data set
- Sampling: obtaining a small sample to **represent** the entire data set ( ---- **undersampling**)
- Oversampling is also required in some scenarios, such as class imbalance problem
  - E.g., 100 HIV test results: 5 positive, 995 negative

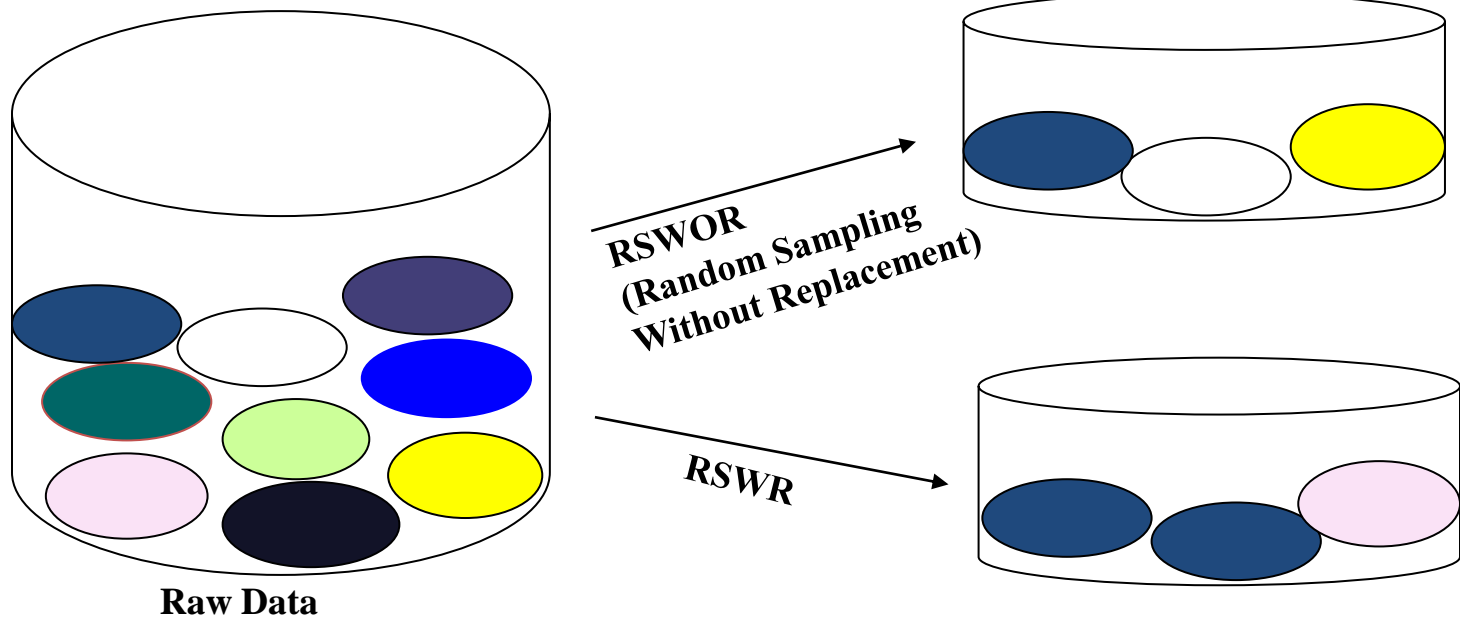
# Sampling Principle

Key principle for effective sampling:

- 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 (1)

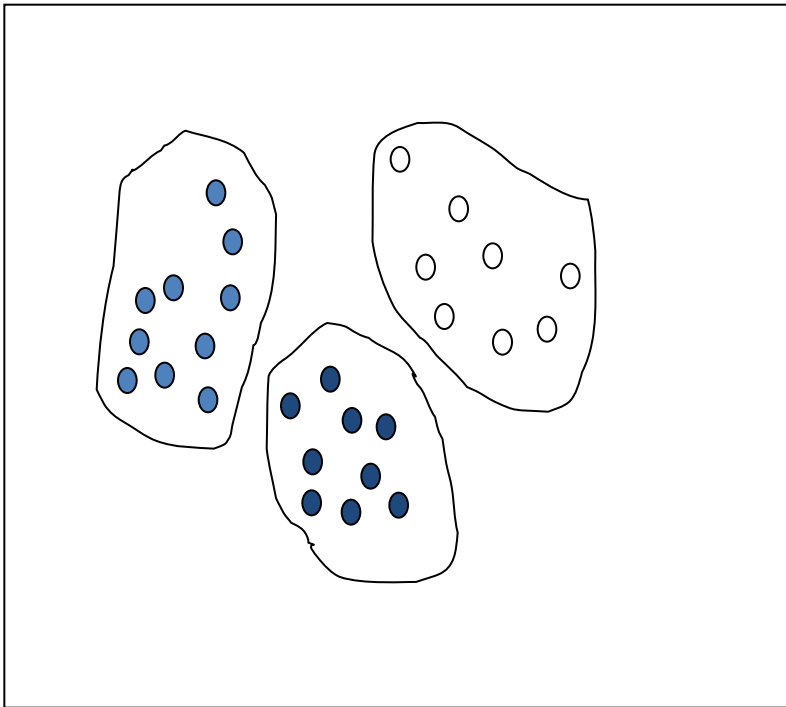
- Random sampling **without replacement**
  - As each example is selected, it is removed from the population
- Random sampling **with replacement**
  - Examples are not removed from the population after being selected
    - The same example can be picked up more than once



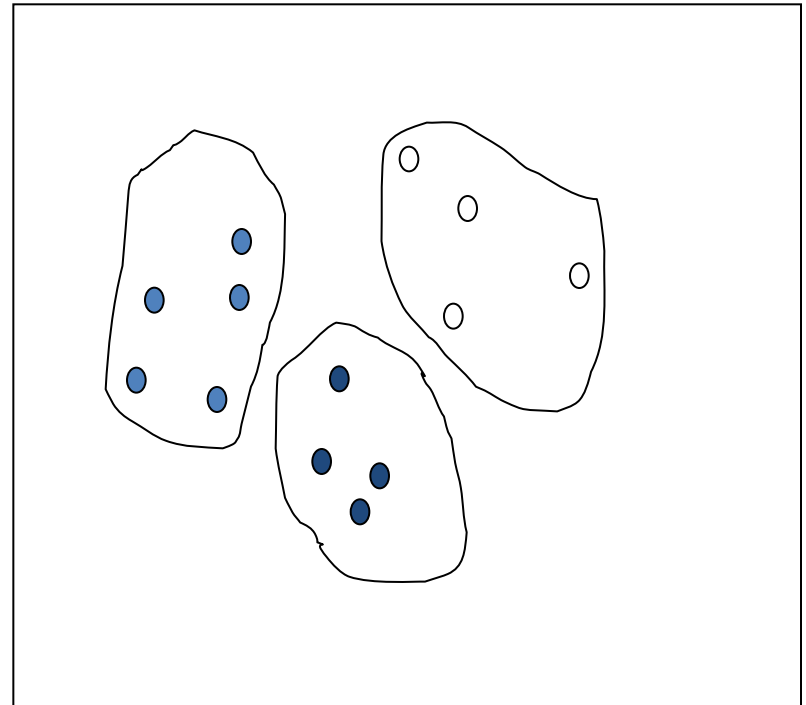


# Types of Sampling (2)

- Stratified sampling
  - Split the data into several partitions; then **draw random samples from each partition**



Raw Data



Stratified Sampling

# Demonstration

- Resample
  - \UCI\waveform-5000
  - Undersampling (with or without replacement)

# Dimensionality Reduction

- Purpose:
  - 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
  - Feature Selection
  - Feature Extraction

# Feature Selection

- Redundant features
  - Duplicated information contained in different features
  - E.g., “Age”, “Year of Birth”; “Purchase price”, “Sales tax”
- Irrelevant features
  - Containing no information that is useful for the task
  - E.g., students' ID is irrelevant to predicting GPA
- Goal:
  - A minimum set of features containing all (most) information

# Heuristic Search in Feature Selection

- Given  $d$  features, there are  $2^d$  possible feature combinations
  - Exhaust search won't work
  - Heuristics has to be applied
- Typical heuristic feature selection methods:
  - Feature ranking
  - Forward feature selection
  - Backward feature elimination
  - Bidirectional search (selection + elimination)
  - Search based on evolution algorithm
  - .....

# Feature Ranking

- Steps:
  - 1) Rank all the individual features according to certain criteria (e.g., information gain, gain ratio,  $\chi^2$ )
  - 2) Select / keep top  $N$  features
- Properties:
  - Usually independent of the learning algorithm to be used
  - Efficient (no search process)
  - Hard to determine the threshold
  - Unable to consider correlation between features

# Forward Feature Selection

- Steps:
  - 1) First select the best single-feature (according to the learning algorithm)
  - 2) Repeat (until some stop criterion is met):  
Select the next best feature, **given the already picked features**
- Properties:
  - Usually learning algorithm **dependent**
  - Feature correlation is considered
  - More reliable
  - Inefficient

# Backward Feature Elimination

- Steps:
  - 1) First build a model based on **all** the features
  - 2) Repeat (until some criterion is met):
    - Eliminate the feature that **makes the least contribution**.
- Properties:
  - Usually learning algorithm **dependent**
  - Feature correlation is considered
  - More reliable
  - Inefficient



# Filter vs Wrapper Model

- Filter model
  - Separating feature selection from learning
  - Relying on general characteristics of data (information, etc.)
  - No bias toward any learning algorithm, fast
  - Feature ranking usually falls into here
- Wrapper model
  - Relying on a predetermined learning algorithm
  - Using predictive accuracy as goodness measure
  - High accuracy, computationally expensive
  - FFS, BFE usually fall into here

# Demonstration

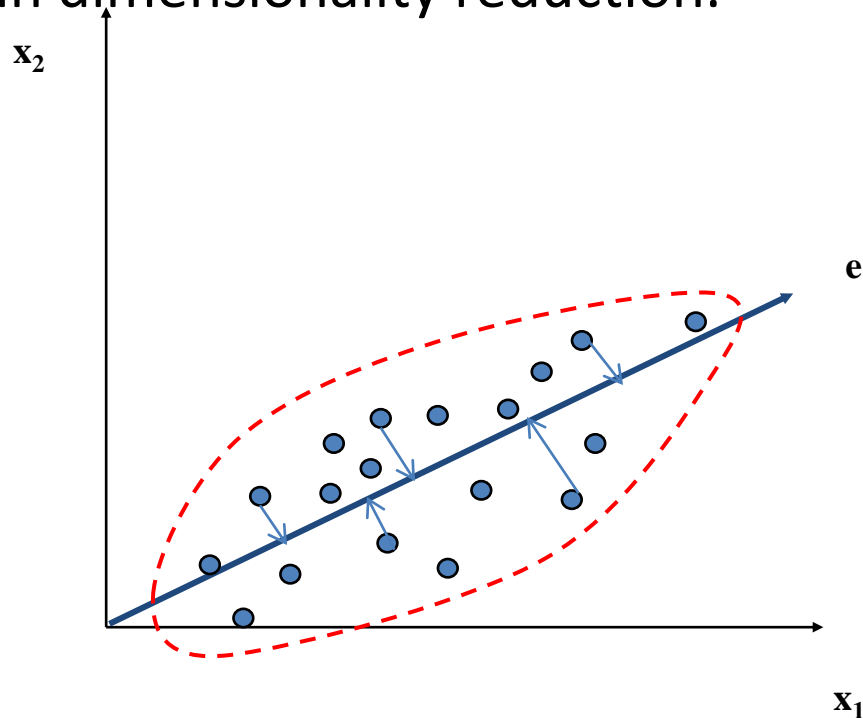
- Feature ranking
  - \Weka\weather
  - ChiSquared, InfoGain, GainRatio
- FFS & BFE
  - \Weka\Diabetes
  - ClassifierSubsetEval + GreedyStepwise

# Feature Extraction

- Map original high-dimensional data onto a lower-dimensional space
  - Generate a (smaller) set of new features
  - Preserve all (most) information from the original data
- Techniques
  - Principal Component Analysis (PCA)
  - Canonical Correlation Analysis (CCA)
  - Linear Discriminant Analysis (LDA)
  - Independent Component Analysis (ICA)
  - Manifold Learning
  - .....

# Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction.



# Principal Component Analysis (Steps)

- Given data from  $n$ -dimensions ( $n$  features), **find  $k \leq n$  new features (*principal components*) that can best represent data**
  - Normalize input data: each feature falls within the same range
  - Compute  $k$  *principal components* (details omitted)
  - Each input data is projected in the new  $k$ -dimensional space
  - The new features (principal components ) are sorted in order of decreasing “significance” or strength
  - Eliminate *weak components / features* to reduce dimensionality.
- **Works for numeric data only**

# PCA Demonstration

- \UCI\breast-w
  - Accuracy with all features
  - PrincipalComponents (data transformation)
  - Visualize/save transformed data (first two features, last two features)
  - Accuracy with all transformed features
  - Accuracy with top 1 or 2 feature(s)

# Outline

- Data
- Data Preprocessing: An Overview
- Data Cleaning
- Data Transformation and Data Discretization
- Data Reduction
- **Summary**

# Summary

- Data (features and instances)
- Data Cleaning: missing values, noise / outliers
- Data Transformation: aggregation, type conversion, normalization
- Data Reduction
  - Sampling: random sampling with replacement, random sampling without replacement, stratified sampling
  - Dimensionality reduction:
    - Feature Selection: Feature ranking, FFS, BFE
    - Feature Extraction: PCA



# Notes

- In real world applications, data preprocessing usually occupies about 70% workload in a data mining task.
- Domain knowledge is usually required to do good data preprocessing.
- To improve a predictive performance of a model
  - Improve learning algorithms (different algorithms, different parameters)
    - Most data mining research focuses on here
  - Improve data quality ---- data preprocessing
    - Deserve more attention!