# Data preprocessing

## Functional Programming and Intelligent Algorithms

Que Tran

Høgskolen i Ålesund

20th March 2017

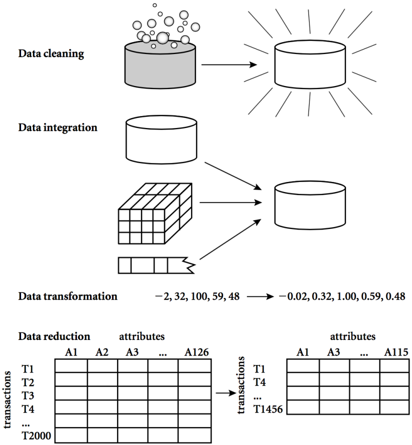


NTNU

# Why data preprocessing?

- Real-world data tend to be dirty
  - incomplete: lacking attribute values, certain attributes of interest, or containing only aggregate data
  - noisy: containing errors, outlier values
  - inconsistent: containing discrepancies in codes
- "How can the data be preprocessed in order to help improve the quality of the data and, consequently, of the mining results?"

# Main tasks in Data preprocessing




*Forms of data preprocessing*

# Data cleaning

Data cleaning attempts to:

— fill in missing values

— smooth noisy data

— identify or remove outliers

— resolve inconsistencies.

# Data cleaning

**Manage missing values:**

- Ignore the instance
- Fill in the missing value manually
- Use a global constant to fill in the missing value
- Use the attribute mean to fill in the missing value
- Use the attribute mean for all instances belonging to the same class as the given instance
- Use the most probable value to fill in the missing value

# Data cleaning

## Noise data

- What is *noise*?
- Manage noise data:
  - Binning
  - Regression
  - Clustering

# Data cleaning

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

**Partition into (equal-frequency) bins:**

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

**Smoothing by bin means:**

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

**Smoothing by bin boundaries:**

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

**Figure 2.11** Binning methods for data smoothing.

# Data cleaning

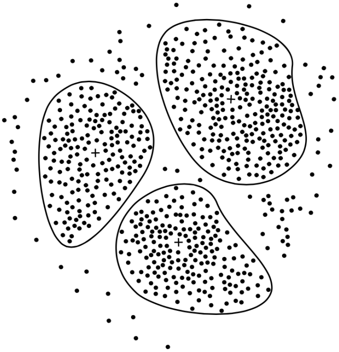


**Figure 2.12** A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster centroid is marked with a "+", representing the average point in space for that cluster. Outliers may be detected as values that fall outside of the sets of clusters.

# Data cleaning

**Manage inconsistent data:**

- Correct inconsistent data manually using external references
- Correct inconsistent data semi-automatically using various tools (Data scrubbing tools, Data auditing tools, Data migration tools...)

# Data integration

- Combines data from multiple sources into a coherent data store
- Some important issues: entity identification problem (schema integration, object matching), redundancy, data value conflicts ...

# Data transformation

- The data are transformed into forms appropriate for mining
- Data transformation involves:
  - Generalization
  - Normalization

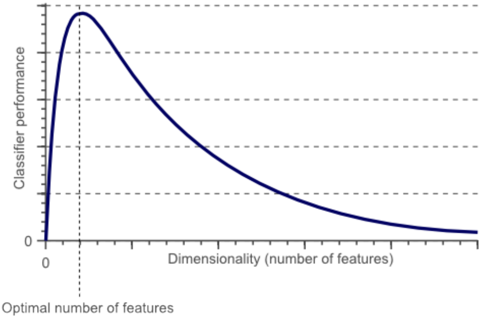
# Data Transformation

- Min-max normalization
  - $v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$
- z-score normalization
  - $v' = \frac{v - \bar{A}}{\sigma_A}$

# Data reduction

- Obtain a reduced representation of the data set that is much smaller in volume, yet produce better or (almost) the same analytical results.
- Why?
  - Computational efficiency
  - Avoid Curse of Dimensionality

# Curse of Dimensionality



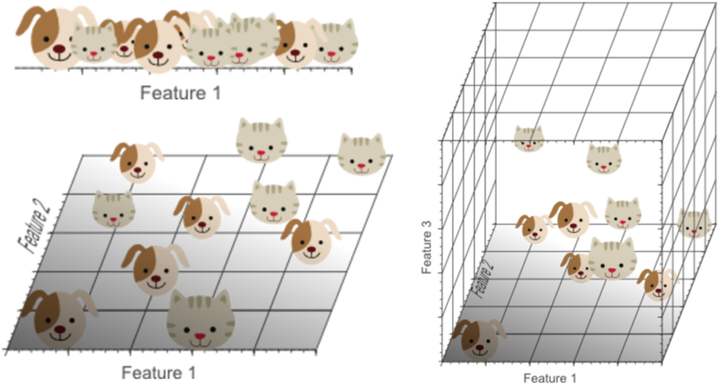

High dimension

— large volume, sparse data

— flexible model

— fits *training data* too well

# Curse of Dimensionality



# Data reduction

Data reduction involves:

— Feature selection

— Feature extraction

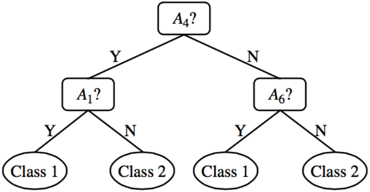
# Data reduction

**Feature selection:**

- Reduces the data set size by removing irrelevant or redundant features.
- Searches for the optimal subset of features
- Feature selection methods are typically greedy
- Basic heuristic methods include the following techniques:
  - Stepwise forward selection
  - Stepwise backward elimination
  - Combination of forward selection and backward elimination
  - Decision tree induction

# Data reduction

## Feature selection:

| Forward selection | Backward elimination | Decision tree induction |
|---|---|---|
| Initial attribute set:<br>$\{A_1, A_2, A_3, A_4, A_5, A_6\}$<br><br>Initial reduced set:<br>{}<br>=> $\{A_1\}$<br>=> $\{A_1, A_4\}$<br>=> Reduced attribute set:<br>$\{A_1, A_4, A_6\}$ | Initial attribute set:<br>$\{A_1, A_2, A_3, A_4, A_5, A_6\}$<br><br>=> $\{A_1, A_3, A_4, A_5, A_6\}$<br>=> $\{A_1, A_4, A_5, A_6\}$<br>=> Reduced attribute set:<br>$\{A_1, A_4, A_6\}$ | Initial attribute set:<br>$\{A_1, A_2, A_3, A_4, A_5, A_6\}$<br><br>=> Reduced attribute set:<br>$\{A_1, A_4, A_6\}$ |



*Greedy (heuristic) methods for attribute subset selection*

# Data reduction

**Feature extraction:**

- Reduces the data set size by transforming feature space to lower dimensional space
- New features do not tell the same meaning as original features
- Data reduction can be *lossless* or *lossy*
- A popular method: Principle Components Analysis (PCA)

## Data reduction

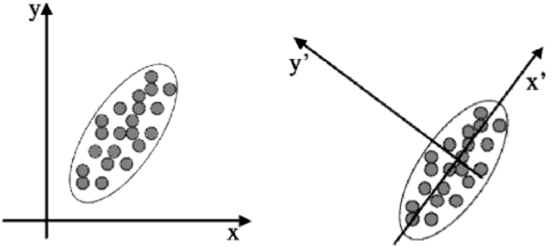



FIGURE 10.6: Two different sets of coordinate axes. The second consists of a rotation and translation of the first and was found using Principal Components Analysis.

# Data reduction

**Principal Component Analysis:**

1. PCA finds a new basis
2. First axis – the principal component
   - ... explains most of the variation
3. Next axis chosen perpendicular to previous axes
   - ... to explain most of the remaining variation

# Data reduction

## PCA Algorithm:

1. Write $N$ data points as rows of a matrix $X$ (size $N \times M$)
2. For each column, subtract its mean to get $B$
3. Compute covariance $C = \frac{1}{N} B^{\mathrm{T}} B$
4. Compute eigenvectors and eigenvalues of $C$
   - $V^{-1} C V = D$
   - $D$: diagonal matrix with eigenvalues
   - $V$: matrix of eigenvectors
5. Sort the columns of $D$ in decreasing order of eigenvalues
   - apply same order to $V$
6. Discard columns with eigenvalue less than $\eta$
7. Transform data by multiplication with $V$