CS 412 Intro. to Data Mining

Chapter 3. Data Preprocessing

Jiawei Han, Computer Science, Univ. Illinois at Urbana-Champaign, 2017

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview

- Data Cleaning

- Data Integration ← การรวมข้อมูลหลาย DB

- Data Reduction and Transformation ← ลดขนาดและแปลงค่าData
  เมื่อก่อนใช้ เพราะ ระบบประมวลยังไม่แรง

- Dimensionality Reduction ← ลด dim ถ้ามีเยอะ ข้อมูลจะกระจาย
  มากเกินไป

- Summary

What is Data Preprocessing? — Major Tasks

- **Data cleaning**
  - Handle missing data, 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

Why Preprocess the Data? — Data Quality Issues

- 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?

ทำให้ข้อมูล
ครบถ้วน และสามารถ
นำไปประมวลผลต่อได้

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
- Data Cleaning or Data Cleansing
- Data Integration
- Data Reduction and Transformation
- Dimensionality Reduction
- Summary

Data Cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, and 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) → Missing ที่แอบอยู่ใน data
    เช่นค่า ดีฟอล ex. ไม่มีใครกรอก
    จะเป็นค่านี้ เช่นถ้าไม่กรอกวันเกิดจะขึ้น 1 มค. จะทำให้เกิดการสับสน
    - Jan. 1 as everyone’s birthday?

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 were not entered due to misunderstanding
  - Certain data may not be considered important at the time of entry
  - Did 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? ← เติมแบบ แมนนวลทีละคน
- 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** สร้างโมเดล จากคอลลัมน์อื่นๆ มาทำนาย ค่าที่หายไป

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**
  - Duplicate records
  - Incomplete data
  - Inconsistent data

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
- Semi-supervised: Combined computer and human inspection
  - Detect suspicious values and check by human (e.g., deal with possible outliers)

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)

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
- Data Cleaning
- Data Integration
- Data Reduction and Transformation
- Dimensionality Reduction
- Summary

Data Integration

- Data integration
  - Combining 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:**
  - 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

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

Dimensionality Reduction

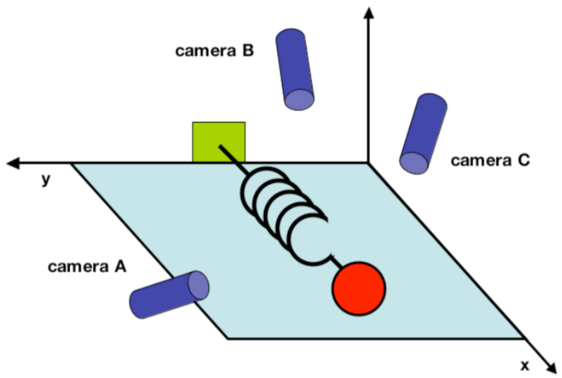
- **Curse of dimensionality**
  - When dimensionality increases, data becomes increasingly sparse
  - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
  - The possible combinations of subspaces will grow exponentially
- **Dimensionality reduction**
  - Reducing the number of random variables under consideration, via obtaining a set of principal variables
- **Advantages of dimensionality reduction**
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization

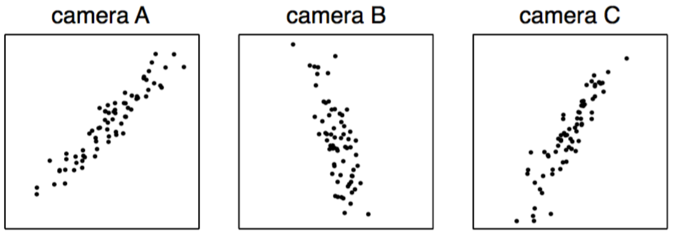
Dimensionality Reduction Techniques

- Dimensionality reduction methodologies
  - **Feature selection**: Find a subset of the original variables (or features, attributes)
  - **Feature extraction**: Transform the data in the high-dimensional space to a space of fewer dimensions
- Some typical dimensionality methods
  - Principal Component Analysis
  - Supervised and nonlinear techniques
    - Feature subset selection
    - Feature creation

Principal Component Analysis (PCA)

- PCA: A statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called ***principal components***
- The original data are projected onto a much smaller space, resulting in dimensionality reduction
- Method: Find the eigenvectors of the covariance matrix, and these eigenvectors define the new space

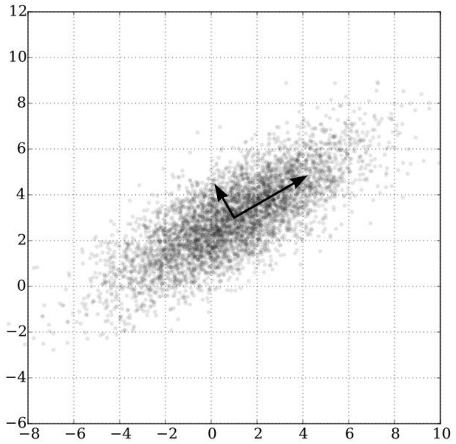




Ball travels in a straight line. Data from three cameras contain much redundancy

Principal Component Analysis (Method)

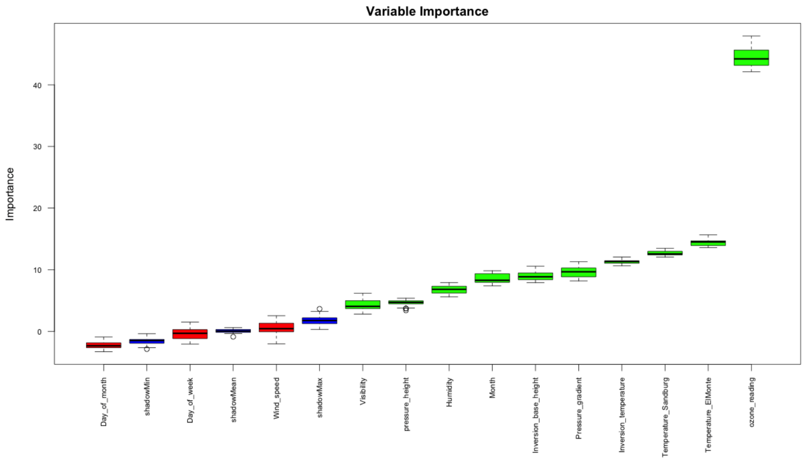
- Given N data vectors from n-dimensions, find $k \leq n$ orthogonal vectors (*principal components*) best used to represent data
  - Normalize input data: Each attribute falls within the same range
  - Compute k orthonormal (unit) vectors, i.e., *principal components*
  - Each input data (vector) is a linear combination of the k principal component vectors
  - The principal components are sorted in order of decreasing "significance" or strength
  - Since the components are sorted, the size of the data can be reduced by eliminating the *weak components*, i.e., those with low variance (i.e., using the strongest principal components, to reconstruct a good approximation of the original data)
- Works for numeric data only



Ack. Wikipedia: Principal Component Analysis

Attribute Subset Selection

- Another way to reduce dimensionality of data
- Redundant attributes
  - Duplicate much or all of the information contained in one or more other attributes
    - E.g., purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
  - Contain no information that is useful for the data mining task at hand
    - Ex. A student's ID is often irrelevant to the task of predicting his/her GPA



Heuristic Search in Attribute Selection

- There are 2^d possible attribute combinations of d attributes
- Typical heuristic attribute selection methods:
  - Best single attribute under the attribute independence assumption: choose by significance tests
  - Best step-wise feature selection:
    - The best single-attribute is picked first
    - Then next best attribute condition to the first, ...
  - Step-wise attribute elimination:
    - Repeatedly eliminate the worst attribute
  - Best combined attribute selection and elimination
  - Optimal branch and bound:
    - Use attribute elimination and backtracking

Attribute Creation (Feature Generation)

- Create new attributes (features) that can capture the important information in a data set more effectively than the original ones
- Three general methodologies
  - Attribute extraction
    - Domain-specific
  - Mapping data to new space (see: data reduction)
    - E.g., Fourier transformation, wavelet transformation, manifold approaches (not covered)
  - Attribute construction
    - Combining features (see: discriminative frequent patterns in Chapter on "Advanced Classification")
    - Data discretization

Summary

- **Data quality**: accuracy, completeness, consistency, timeliness, believability, interpretability
- **Data cleaning**: e.g. missing/noisy values, outliers
- **Data integration** from multiple sources:
  - Entity identification problem; Remove redundancies; Detect inconsistencies
- **Data reduction, data transformation and data discretization**
  - Numerosity reduction; Data compression
  - Normalization; Concept hierarchy generation
- **Dimensionality reduction**

References

- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Comm. of ACM, 42:73-78, 1999
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley, 2003
- T. Dasu, T. Johnson, S. Muthukrishnan, V. Shkapenyuk. Mining Database Structure; Or, How to Build a Data Quality Browser. SIGMOD'02
- H. V. Jagadish et al., Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), Dec. 1997
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999
- E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. *IEEE Bulletin of the Technical Committee on Data Engineering. Vol.23, No.4*
- V. Raman and J. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB'2001
- T. Redman. Data Quality: Management and Technology. Bantam Books, 1992
- R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. IEEE Trans. Knowledge and Data Engineering, 7:623-640, 1995