Lecture 1.

1). Knowledge Discovery (KDD) Process.

- a). SEMMA ~~Data~~ Mining Model ⇒ P2
- b). CRISP-DM Data Mining Model

2). Data Warehousing: 4 property. ⇒ P5

- a). Enterprise warehouse
- b). Data Mart ⇒ Extraction, Transformation (ETL), Loading
- c). Virtual warehouse

*: Star schema, Snowflake schema, Fact constellations. ⇒ P7

3). On-line Analytical Processing (OLAP) ⇒ P8.

Lecture 2.

1). Attribute Types. ⇒ P10

2). Basic Statistical Descriptions of Data

3). Data Visualization ⇒ P12-13

Lecture 3.

1). Data Quality.

2). Data Cleaning ⇒ P16-17
- Missing value
- Noisy data
- Redundancy

3). Correlation Analysis (Numerical Data) — $(r_{A,B})$ ⇒ P17-18

4). Correlation Aalysis (Categorical Data). — χ^2

5). Data Reduction Strategies

6). Data Transformation. ⇒ P19
- Min-Max normalization
- Z-score normalization

7). Binning ⇒ P19.
- Equal-width.
- Equal-depth.

Lecture 4.

1). ~~Holdout~~ Method, Cross-validation, Bootstrap ⇒ P21

2). Difference between classification and Prediction ⇒ P21-22.

3). ⇒ P22.
- Assumptions of Linear Regression.
- variable selection in Linear Regression.

4). R^2 & Adjusted R^2.

Lecture 5.

1) Principal Components Analysis (PCA)

2). ROC Curves, Gain Chart, Lift chart. ⇒ P26-28

Lecture 6. ←2) Supervised vs. Unsupervised Learning.

1). Decision Tree (summary) ⇒ P30

a). Info Gain / Gain Ratio ⇒ P31

b). Gini Index ⇒ P32

Lecture 7.

1) Ensemble Methods

a). Bagging : Bootstrap Aggragation

b). Boosting : Adaboost

C). Random Forest : Forest-RI, Forest-RC

2). Class-Imbalanced Data Sets.

Lecture 8

Neural Network

Lecture 9.

1). Frequent pattern

2). support, confidence, expected confidence, lift.

3). Closed Patterns, Max-Patterns

4). Apriori pruning principle

Addition :

1). A and D of PCA

2). A and D of Decision Tree with more tree depth

3). The same and difference of Validation and Testing

4). for numic input, ordinal is better than interval

two of

} ⇒ P33

Lecture 1. Introduction to Data Mining.

ISE 365/465 –Applied Data Mining – Overview, Ch 1. and 4.1-2

Mike Magent
Industrial and Systems Engineering Dept.
Lehigh University
Spring 2016

Class Agenda

- Class Roster
- Who am I?
- Course Coverage
- Syllabus
- Introduction to Data Mining
  - What is Data Mining?
  - CRISP-DM

Who am I?

- Graduated from Penn State (B.S.) and Lehigh (M.S. and Ph.D.) Industrial Engineering Departments
- Data & Advanced Analytics and Predictive Modeling Lead in Decision Sciences Department at Air Products
  - Worked in Decision Sciences at AP since 1996
- Previously taught Deterministic OR, Advanced OR, and Supply Chain courses in this department

Air Products Overview

- Airproducts.com
- Industrial Gases and Chemicals Company
- Located in Trexlertown west of Allentown
- Tasks I have worked on in Decision Sciences
  - Vehicle Routing
  - Simulation
  - SAP APO – Production Planning and Scheduling
  - Sales and Operations Planning
  - Data Mining

Why Data Mining?

- The Explosive Growth of Data: from terabytes to petabytes
  - kilo-10^3 mega-10^6 giga-10^9 tera-10^{12} peta-10^{15} exa-10^{18}
  - Data collection and data availability
    - Automated data collection tools, database systems, Web, computerized society
  - Major sources of abundant data
    - Business: Web, e-commerce, transactions, stocks, ...
    - Science: Remote sensing, bioinformatics, scientific simulation, ...
    - Society and everyone: news, digital cameras, YouTube

- We are drowning in data, but starving for knowledge!
- "Necessity is the mother of invention"—Data mining—Automated analysis of massive data sets

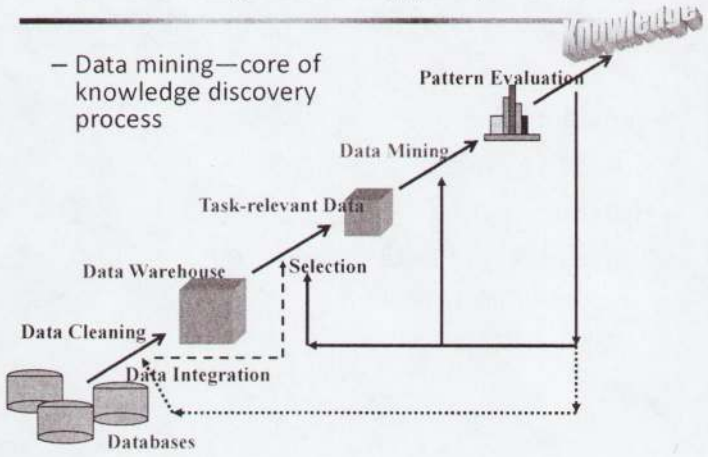
What Is Data Mining?

- Data mining (knowledge discovery from data)
  - Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data
- Alternative names
  - **Advanced Analytics, Predictive Modeling, Predictive Analytics,** Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, Big Data, etc.
- Watch out: Is everything "data mining"?
  - Simple search and query processing
  - (Deductive) expert systems
  - Leads to confusion – Everyone thinks they can data mine

KDD Process.

Knowledge Discovery (KDD) Process

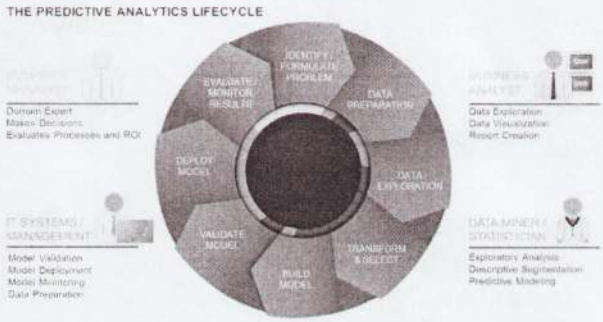
– Data mining—core of knowledge discovery process



McKinsey Big Data and Advanced Analytics Video

- http://www.mckinsey.com/insights/marketing_sales/putting_big_data_and_advanced_analytics_to_work

SAS Predictive Analytics Lifecycle



model 1.

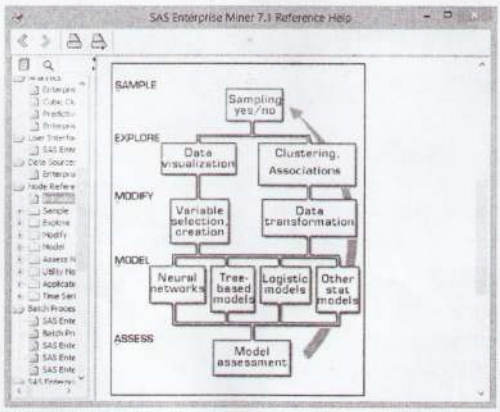
SEMMA Data Mining Model

- **Sample**. The process starts with data sampling, e.g., selecting the data set for modeling. The data set should be large enough to contain sufficient information to retrieve, yet small enough to be used efficiently. This phase also deals with data partitioning.
- **Explore**. This phase covers the understanding of the data by discovering anticipated and unanticipated relationships between the variables, and also abnormalities, with the help of data visualization.
- **Modify**. The Modify phase contains methods to select, create and transform variables in preparation for data modeling.
- **Model**. In the Model phase the focus is on applying various modeling (data mining) techniques on the prepared variables in order to create models that possibly provide the desired outcome.
- **Assess**. The last phase is Assess. The evaluation of the modeling results shows the reliability and usefulness of the created models.

Source: SAS Enterprise Miner website

Source: SAS Enterprise Miner help

SAS SEMMA Process for Data Mining



Source: SAS Enterprise Miner help

model 2.

(CRoss-Industry Standard Process for Data Mining)

CRISP-DM Data Mining Model

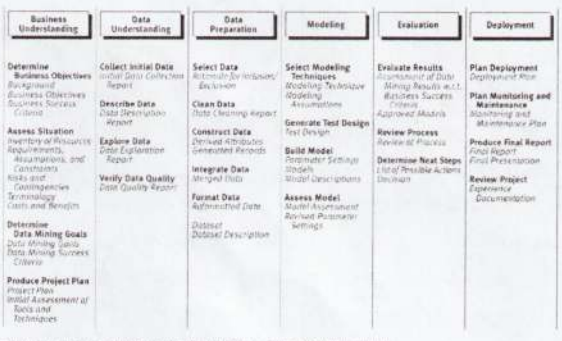


Figure 3: Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model

Source: CRISP-DM 1.0 Guide from SPSS.com. Also see www.crisp-dm.org

Foundational Principles

Foundational Principles for Applied Data Mining

- People
  - Soft Skills to transform clients to support analytics
- Data
  - Quality and understanding
- Simplicity
  - Create the simplest solution to solve the problem well
  - "Don't let perfect be the enemy of good"
  - Complexity is harder to explain, deploy, and support

People Issues in Data Mining

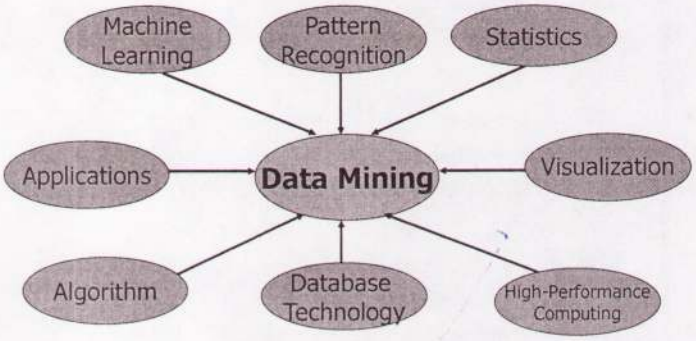
- Following SEMMA/CRISP-DM gives structure, but does not address all people issues for a DM project
  - Does management want the analysis?
  - Who will champion the project?
  - Who understands the business problem?
  - Who understands the data?
  - Who will use the results and how can the results be deployed to them?
  - What is the background of people on the project?
  - How to communicate results to clients?
  - Video...

14

People - Resources for Soft Skills

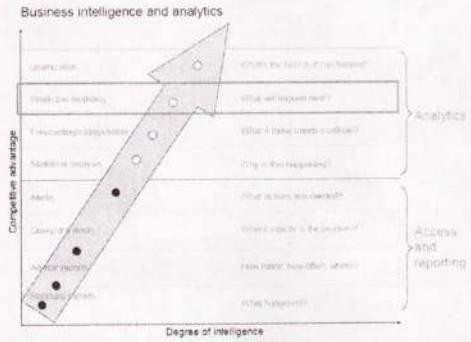
- That '70s Show Video – Career Day
- Data-informed.com
  - "Why Soft Skills Matter in Data Science" Article
- SAS
  - "Why people and process matter, in addition to great technology, in predictive analytics"
- Analytics Magazine
  - Soft skills: The 'killer app' for analytics

Data Mining: Confluence of Multiple Disciplines



16

Business Intelligence and Analytics



From "Competing on Analytics" by Davenport and Harris, 2007 and http://contentperspective.se/

Why Not Traditional Data Analysis?

etc.

- Tremendous amount of data
  - Algorithms must be highly scalable to handle such as tera-bytes of data
- High-dimensionality of data
  - Micro-array may have tens of thousands of dimensions
- High complexity of data
  - Data streams and sensor data
  - Time-series data, temporal data, sequence data
  - Structure data, graphs, social networks and multi-linked data
  - Heterogeneous databases and legacy databases
  - Spatial, spatiotemporal, multimedia, text and Web data
  - Software programs, scientific simulations
- New and sophisticated applications

18

Multi-Dimensional View of Data Mining

- Data to be mined
  - Relational, data warehouse, transactional, stream, object-oriented/relational, active, spatial, time-series, text, multi-media, heterogeneous, legacy, WWW
- Knowledge to be mined
  - Characterization, discrimination, association, classification, clustering, trend/deviation, outlier analysis, etc.
  - Multiple/integrated functions and mining at multiple levels
- Techniques utilized
  - Database-oriented, data warehouse (OLAP), machine learning, statistics, visualization, etc.
- Applications adapted
  - Retail, telecommunication, banking, fraud analysis, bio-data mining, stock market analysis, text mining, Web mining, etc.

19

Data Mining: On What Kinds of Data?

- Database-oriented data sets and applications
  - Relational database, data warehouse, transactional database
- Advanced data sets and advanced applications
  - Data streams and sensor data
  - Time-series data, temporal data, sequence data (incl. bio-sequences)
  - Structure data, graphs, social networks and multi-linked data
  - Object-relational databases
  - Heterogeneous databases and legacy databases
  - Spatial data and spatiotemporal data
  - Multimedia database
  - Text databases
  - The World-Wide Web

20

ISE 365/465 Coverage (Chapters 1-4, 6, 8-10)

- ISE 365/465 Coverage
  - Introduction
  - Data Preprocessing
  - Data Warehouse and OLAP Technology: An Introduction
  - Classification and Prediction
  - Mining Frequent Patterns, Association and Correlations
  - Cluster Analysis
  - Real-world application of the above
  - Demos, assignments, and projects SAS Enterprise Guide and Enterprise Miner Software

21

Where to Find References? DBLP, CiteSeer, Google

- Data mining and KDD (SIGKDD: CDROM)
  - Conferences: ACM-SIGKDD, IEEE-ICDM, SIAM-DM, PKDD, PAKDD, etc.
  - Journal: Data Mining and Knowledge Discovery, KDD Explorations, ACM TKDD
- Database systems (SIGMOD: ACM SIGMOD Anthology—CD ROM)
  - Conferences: ACM-SIGMOD, ACM-PODS, VLDB, IEEE-ICDE, EDBT, ICDT, DASFAA
  - Journals: IEEE-TKDE, ACM-TODS/TOIS, JIIS, J. ACM, VLDB J., Info. Sys., etc.
- AI & Machine Learning
  - Conferences: Machine learning (ML), AAAI, IJCAI, COLT (Learning Theory), CVPR, NIPS, etc.
  - Journals: Machine Learning, Artificial Intelligence, Knowledge and Information Systems, IEEE-PAMI, etc.
- Web and IR
  - Conferences: SIGIR, WWW, CIKM, etc.
  - Journals: WWW: Internet and Web Information Systems,
- Statistics
  - Conferences: Joint Stat. Meeting, etc.
  - Journals: Annals of statistics, etc.
- Visualization
  - Conference proceedings: CHI, ACM-SIGGraph, etc.
  - Journals: IEEE Trans. visualization and computer graphics, etc.

22

Data Mining Software

- Enterprise Data Mining Software (for a Cost)
  - SAS Enterprise Guide and Enterprise Miner
    - We will use these in this course
  - IBM SPSS Modeler
  - Several Others
- Open Source Data Mining Software (Free)
  - R (Rattle GUI available for non-coders)
  - KNIME
  - RapidMiner
  - Weka

23

Introduction Summary

- Data mining: Discovering interesting patterns from large amounts of data
- A natural evolution of database technology, in great demand, with wide applications
- CRISP-DM and SEMMA processes provide roadmap for a data mining project
- 3 foundational principles of data mining:
  - People
  - Data
  - Simplicity
- Mining can be performed in a variety of information repositories
- Data mining functionalities: characterization, discrimination, association, classification, clustering, outlier and trend analysis, etc.

24

Chapter 4.1-2: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP

What is a Data Warehouse?

- Defined in many different ways, but not rigorously.
  - A decision support database that is maintained separately from the organization's operational database
  - Support information processing by providing a solid platform of consolidated, historical data for analysis.
- "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."—W. H. Inmon
- Data warehousing:
  - The process of constructing and using data warehouses

Data Warehouse—Subject-Oriented

- Organized around major subjects, such as customer, product, sales
- Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing
- Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process

Data Warehouse—Integrated

- Constructed by integrating multiple, heterogeneous data sources
  - relational databases, flat files, on-line transaction records
- Data cleaning and data integration techniques are applied.
  - Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
    - E.g., Hotel price: currency, tax, breakfast covered, etc.
  - When data is moved to the warehouse, it is converted.

Data Warehouse—Time Variant

- The time horizon for the data warehouse is significantly longer than that of operational systems
  - Operational database: current value data
  - Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)
- Every key structure in the data warehouse
  - Contains an element of time, explicitly or implicitly
  - But the key of operational data may or may not contain "time element"

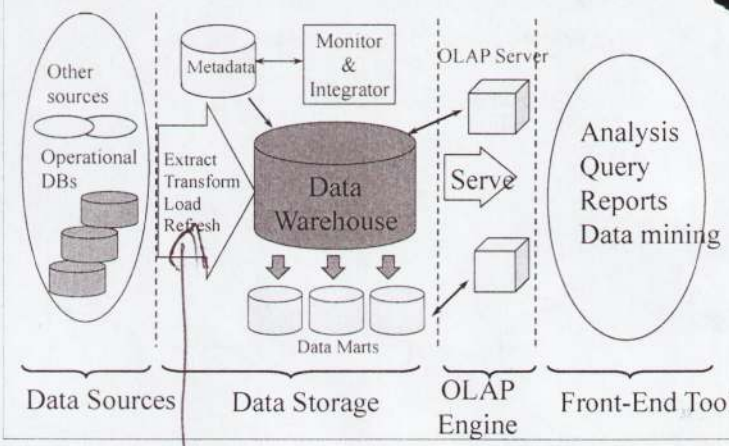
Data Warehouse—Nonvolatile

- A physically separate store of data transformed from the operational environment
- Operational update of data does not occur in the data warehouse environment
  - Does not require transaction processing, recovery, and concurrency control mechanisms
  - Requires only two operations in data accessing:
    - *initial loading of data* and *access of data*

Why a Separate Data Warehouse?

- High performance for both systems
  - DBMS—tuned for OLTP: access methods, indexing, concurrency control, recovery
  - Warehouse—tuned for OLAP: complex OLAP queries, multidimensional view, consolidation
- Different functions and different data:
  - missing data: Decision support requires historical data which operational DBs do not typically maintain
  - data consolidation: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
  - data quality: different sources typically use inconsistent data representations, codes and formats which have to be reconciled
- Note: There are more and more systems which perform OLAP analysis directly on relational databases

Data Warehouse: A Multi-Tiered Architecture



Three Data Warehouse Models

- Enterprise warehouse
  - collects all of the information about subjects spanning the entire organization
- Data Mart
  - a subset of corporate-wide data that is of value to a specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart
    - Independent vs. dependent (directly from warehouse) data mart
- Virtual warehouse
  - A set of views over operational databases
  - Only some of the possible summary views may be materialized

Extraction, Transformation, and Loading (ETL)

- **Data extraction**
  - get data from multiple, heterogeneous, and external sources
- **Data cleaning**
  - detect errors in the data and rectify them when possible
- **Data transformation**
  - convert data from legacy or host format to warehouse format
- **Load**
  - sort, summarize, consolidate, compute views, check integrity, and build indicies and partitions
- **Refresh**
  - propagate the updates from the data sources to the warehouse

Metadata Repository

- **Meta data** is the data defining warehouse objects. It stores:
- Description of the structure of the data warehouse
  - schema, view, dimensions, hierarchies, derived data defn, data mart locations and contents
- Operational meta-data
  - data lineage (history of migrated data and transformation path), currency of data (active, archived, or purged), monitoring information (warehouse usage statistics, error reports, audit trails)
- The algorithms used for summarization
- The mapping from operational environment to the data warehouse
- Data related to system performance
  - warehouse schema, view and derived data definitions
- Business data
  - business terms and definitions, ownership of data, charging policies

Chapter 4.1-2: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP

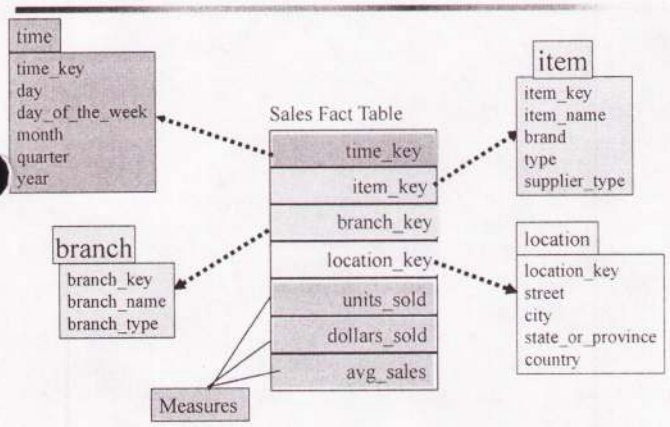
From Tables and Spreadsheets to Data Cubes

- A **data warehouse** is based on a multidimensional data model which views data in the form of a data cube
- A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions
  - **Dimension tables**, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
  - **Fact table** contains **measures** (such as dollars_sold) and keys to each of the related dimension tables
- In data warehousing literature, an n-D base cube is called a base cuboid. The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid. The lattice of cuboids forms a data cube.

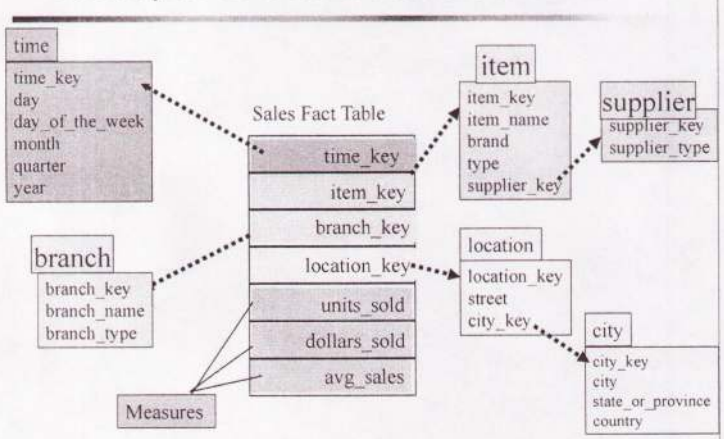
Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - Star schema: A fact table in the middle connected to a set of dimension tables
  - Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

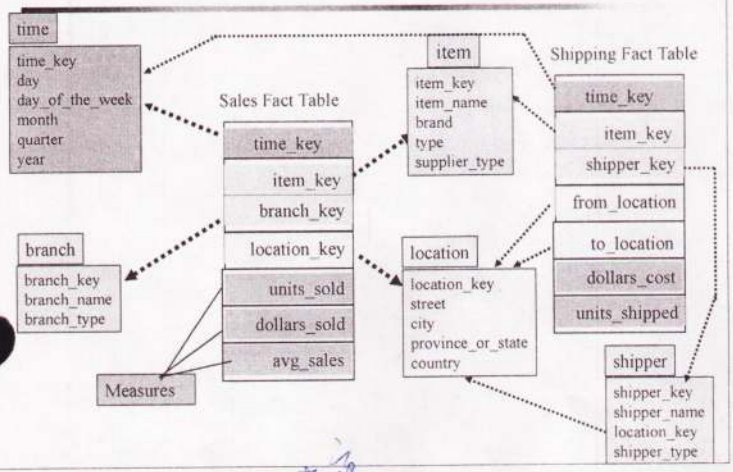
Example of **Star Schema**



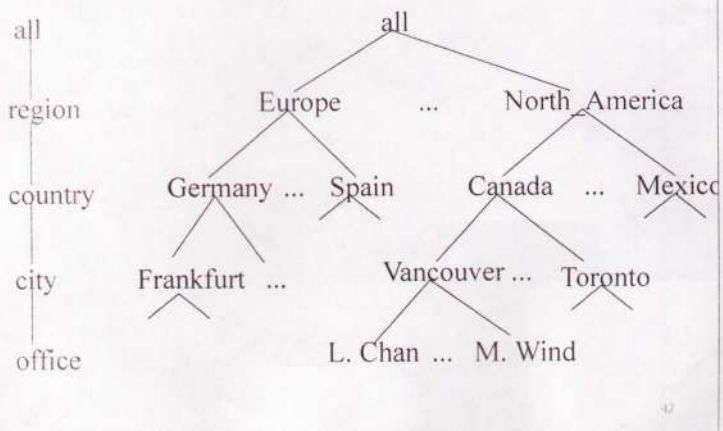
Example of **Snowflake Schema**



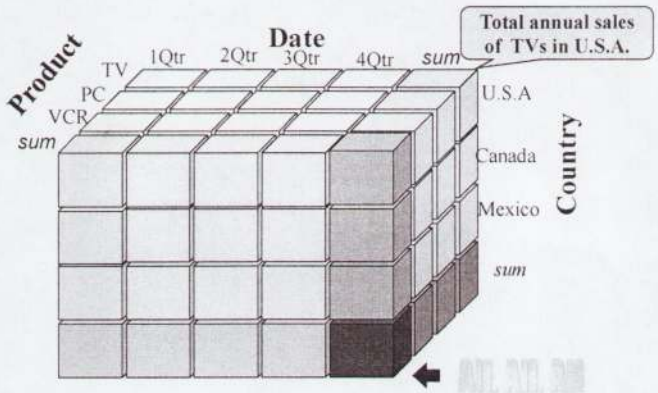
Example of **Fact Constellation**



A Concept Hierarchy: **Dimension** (location)



A Sample Data Cube



Typical OLAP Operations

- Roll up (drill-up): summarize data
  - *by climbing up hierarchy or by dimension reduction*
- Drill down (roll down): reverse of roll-up
  - *from higher level summary to lower level summary or detailed data, or introducing new dimensions*
- Slice and dice: *project and select*
- Pivot (rotate):
  - *reorient the cube, visualization, 3D to series of 2D planes*
- Other operations
  - *drill across: involving (across) more than one fact table*
  - *drill through: through the bottom level of the cube to its back-end relational tables (using SQL)*

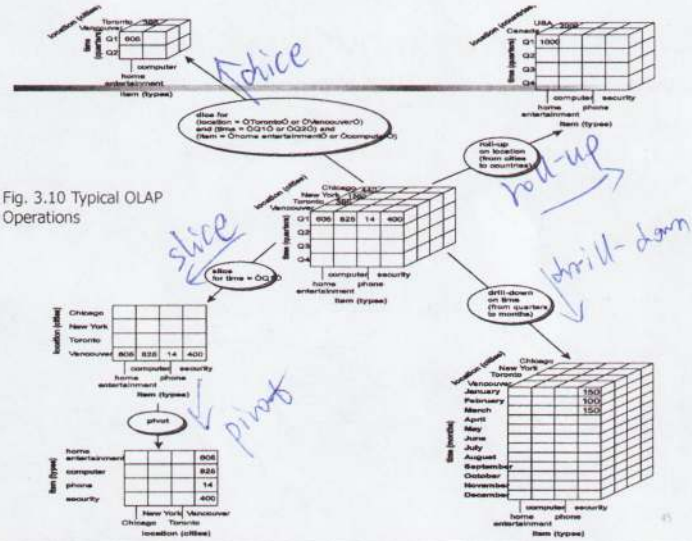
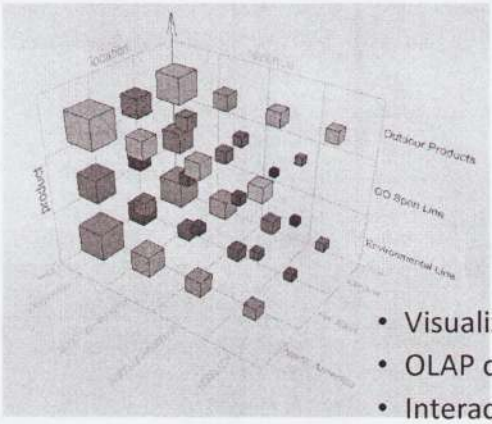


Fig. 3.10 Typical OLAP Operations

Browsing a Data Cube



- Visualization
- OLAP capabilities
- Interactive manipulation

Lecture 2 Know your Data and Data Preprocessing

Class Agenda

- Introduction to Modeler Stream Building
  - Source Nodes – How to Read Data into Modeler
  - Type Node, Filter Node, Derive Node, Filler Node, Select Node, Sort Node, Field Reorder Node, Table Node
- Data Preprocessing
  - Data Summarization
    - Aggregation Node
  - Data Cleaning
  - Visual Analysis of Data
  - Modeler Example

Introduction to Modeling in Modeler

- Modeler Models are created using streams
- Working with SPSS Modeler is a three-step process of working with data.
  - First, you read data into SPSS Modeler,
  - Then, run the data through a series of manipulations,
  - And finally, send the data to a destination.
- User's Guide in help has a lot of info
  - Now we will look at the Modeler GUI

Different Types of Nodes in Modeler

- • **Sources.** Nodes bring data into SPSS Modeler.
- • **Record Ops.** Nodes perform operations on data **records**, such as selecting, merging, and appending.
- • **Field Ops.** Nodes perform operations on data **fields**, such as filtering, deriving new fields, and determining the measurement level for given fields.
- • **Graphs.** Nodes graphically display data before and after modeling. Graphs include plots, histograms, web nodes, and evaluation charts.
- • **Modeling.** Nodes use the modeling algorithms available in SPSS Modeler, such as neural nets, decision trees, clustering algorithms, and data sequencing.
- • **Database Modeling.** Nodes use the modeling algorithms available in Microsoft SQL Server, IBM DB2, and Oracle databases. - We don't have this
- • **Output.** Nodes produce a variety of output for data, charts, and model results that can be viewed in SPSS Modeler.
- • **Export.** Nodes produce a variety of output that can be viewed in external applications, such as IBM® SPSS® Data Collection or Excel.
- • **SPSS Statistics.** Nodes import data from, or export data to, IBM® SPSS® Statistics, as well as running SPSS Statistics procedures.

Source: Modeler help

Accessing Sources of Data in Modeler

- Databases
- Text Files
  - Variable Length
  - Fixed Length
- Spreadsheets
- Statistics files (SPSS, SAS)
- XML Files
- User input

(CRoss-Industry Standard Process for Data Mining)

CRISP-DM Data Mining Model

We are discussing this area

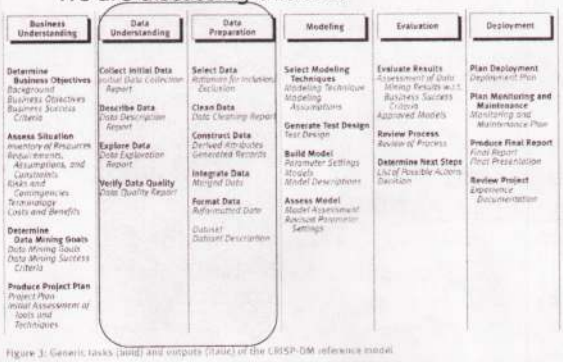


Figure 3: Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model

Source: CRISP-DM 1.0 Guide from SPSS.com. Also see www.crisp-dm.org

Chapter 2: Getting to Know Your Data

- Data Objects and Attribute Types
- Basic Statistical Descriptions of Data
- Data Visualization
- Measuring Data Similarity and Dissimilarity
- Summary

Types of Data Sets

- Record
  - Relational records
  - Data matrix, e.g., numerical matrix, crosstabs
  - Document data: text documents: term-frequency vector
  - Transaction data
- Graph and network
  - World Wide Web
  - Social or information networks
  - Molecular Structures
- Ordered
  - Video data: sequence of images
  - Temporal data: time-series
  - Sequential Data: transaction sequences
  - Genetic sequence data
- Spatial, image and multimedia:
  - Spatial data: maps
  - Image data:
  - Video data:

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

7

Data Objects

- Data sets are made up of 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 *samples , examples, instances, data points, objects, tuples.*
- Data objects are described by **attributes.**
- Database rows -> data objects; columns ->attributes.

8

Attributes

- **Attribute (**or **dimensions, features, variables**): a data field, representing a characteristic or feature of a data object.
  - *E.g., customer _ID, name, address*
- Types:
  - Nominal
  - Binary
  - Numeric: quantitative
    - Interval-scaled
    - Ratio-scaled

9

Categorical Attribute Types

- **Nominal:** categories, states, or "names of things"
  - *Hair_color = {auburn, black, blond, brown, grey, red, white}*
  - marital status, occupation, ID numbers, zip codes
- **Binary**
  - Nominal attribute with only 2 states (0 and 1)
  - Symmetric binary: both outcomes equally important
    - e.g., gender
  - Asymmetric binary: outcomes not equally important.
    - e.g., medical test (positive vs. negative)
    - Convention: assign 1 to most important outcome (e.g., HIV positive)
- **Ordinal**
  - Values have a meaningful order (ranking) but magnitude between successive values is not known.
  - *Size = {small, medium, large}*, grades, army rankings

10

Numeric Attribute Types

- Quantity (integer or real-valued)
- **Interval**
  - Measured on a scale of **equal-sized units**
  - Values have order
    E.g., *temperature in C°or F°, calendar dates*
  - No true zero-point
- **Ratio**
  - Inherent **zero-point**
  - We can speak of values as being an order of magnitude larger than the unit of measurement (10 K° is twice as high as 5 K°).
    e.g., *temperature in Kelvin, length, counts, monetary quantities*

11

Discrete vs. Continuous Attributes

- **Discrete Attribute**
  - Has only a finite or countably infinite set of values
    - E.g., zip codes, profession, or the set of words in a collection of documents
  - Sometimes, represented as integer variables
  - Note: Binary attributes are a special case of discrete attributes
- **Continuous Attribute**
  - Has real numbers as attribute values
    - E.g., temperature, height, or weight
  - Practically, real values can only be measured and represented using a finite number of digits
  - Continuous attributes are typically represented as floating-point variables

12

Data Types in Modeler

The following measurement levels are available:

- **Default.** Data whose storage type and values are unknown (for example, because they have not yet been read) are displayed as **<Default>**.
- **Continuous.** Used to describe numeric values, such as a range of 0–100 or 0.75–1.25. A continuous value can be an integer, real number, or date/time.
- **Categorical.** Used for string values when an exact number of distinct values is unknown. This is an **uninstantiated** data type, meaning that all possible information about the storage and usage of the data is not yet known. Once data have been read, the measurement level will be *Flag*, *Nominal*, or *Typeless*, depending on the maximum set size specified in the Stream Properties dialog box.
- **Flag.** Used for data with two distinct values that indicate the presence or absence of a trait, such as true and false, Yes and No or 0 and 1. The values used may vary, but one must always be designated as the "true" value, and the other as the "false" value. Data may be represented as text, integer, real number, date, time, or timestamp.
- **Nominal.** Used to describe data with multiple distinct values, each treated as a member of a set, such as small/medium/large. Nominal data can have any storage—numeric, string, or date/time. Note that setting the measurement level to *Nominal* does not automatically change the values to string storage.
- **Ordinal.** Used to describe data with multiple distinct values that have an inherent order. For example, salary categories or satisfaction rankings can be typed as ordinal data. The order is defined by the natural sort order of the data elements. For example, 1, 3, 5 is the default sort order for a set of integers, while HIGH, LOW, NORMAL (ascending alphabetically) is the order for a set of strings. The ordinal measurement level enables you to define a set of categorical data as ordinal data for the purposes of visualization, model building, and export to other applications (such as IBM® SPSS® Statistics) that recognize ordinal data as a distinct type. You can use an ordinal field anywhere that a nominal field can be used. Additionally, fields of any storage type (real, integer, string, date, time, and so on) can be defined as ordinal.
- **Typeless.** Used for data that does not conform to any of the above types, for fields with a single value, or for nominal data where the set has more members than the defined maximum. It is also useful for cases in which the measurement level would otherwise be a set with many members (such as an account number). When you select **Typeless** for a field, the role is automatically set to **None**, with **Record ID** as the only alternative. The default maximum size for sets is 250 unique values. This number can be adjusted or disabled on the Options tab of the Stream Properties dialog box, which can be accessed from the Tools menu.

Source: Modeler help

Introduction to Modeling in Modeler

- Simple Example in Demo Folder: ModelingIntro stream
  - Will show use of Type Node, Filter Node, Select Node, Sort Node, Field Reorder Node, Table Node
  - These are nodes that will be the building blocks for your Modeler Streams

Chapter 2: Getting to Know Your Data

- Data Objects and Attribute Types
- Basic Statistical Descriptions of Data
- Data Visualization
- Measuring Data Similarity and Dissimilarity
- Summary

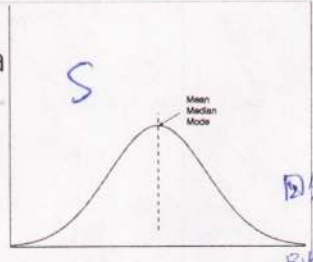
Measuring the Central Tendency

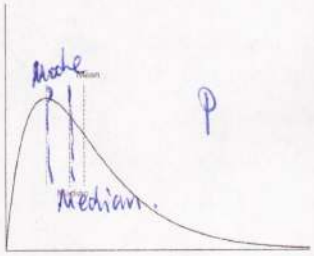
- Mean (algebraic measure) (sample vs. population): $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \mu = \frac{\sum x}{N}$
  - Weighted arithmetic mean: $\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$
  - Trimmed mean: chopping extreme values
- Median: A holistic measure
  - Middle value if odd number of values, or average of the middle two values otherwise
- Mode
  - Value that occurs most frequently in the data
  - Unimodal, bimodal, trimodal
  - Empirical formula:

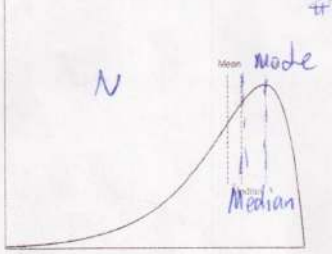
$$mean - mode = 3 \times (mean - median)$$

Symmetric vs. Skewed Data

- Median, mean and mode of symmetric, positively and negatively skewed data







Measuring Dispersion - Data Audit Node in Modeler

- **Data Audit Node (will see today in Modeler)**- outliers by Standard Deviation or Quartile
  - Quartiles: Q_1 (25th percentile), Q_3 (75th percentile)
  - Inter-quartile range: $IQR = Q_3 - Q_1$
  - Five number summary: min, Q_1, Mean, Q_3, max
  - Outlier: a value higher/lower than 1.5 x IQR
  - Skewness, Uniques Values, Valid Values
- Variance and standard deviation (*sample: s, population: σ*)
  - Variance: (algebraic, scalable computation)

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right] \qquad \sigma^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2 = \frac{1}{N}\sum_{i=1}^{n} x_i^2 - \mu^2$$

  - Standard deviation *s (or σ)* is the square root of variance s^2 (or σ^2)

Data Cleaning

- Importance
  - "Data cleaning is one of the three biggest problems in data warehousing"—Ralph Kimball
  - "Data cleaning is the number one problem in data warehousing"—DCI survey
- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data
  - Resolve redundancy caused by data integration

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 (assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably.
- Fill in the missing value manually: tedious + infeasible?
- **Data Audit Node -** Fill in it automatically with
  1. a global constant : e.g., "unknown", a new class?!
  2. the attribute mean
  3. 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

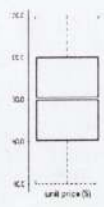
Modeler Data Summarization Examples

- Telco_dataaudit stream in Demo folder
  - Will show the use of the data audit node
  - Will also show filler and derive nodes
- ADP_basic_demo stream in Demo folder
  - Shows Modeler's auto data prep node
    - **This node can be dangerous! – It does a lot of things (some of which are questionable) with very little explanation**

Chapter 2: Getting to Know Your Data

- Data Objects and Attribute Types
- Basic Statistical Descriptions of Data
- Data Visualization
- Measuring Data Similarity and Dissimilarity
- Summary

Boxplot Analysis

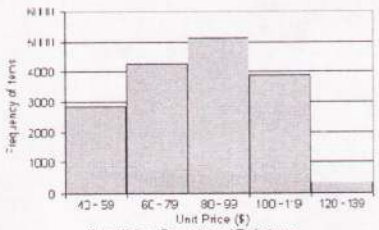


- Five-number summary of a distribution:
  Minimum, Q1, M, Q3, Maximum
- Boxplot
  - Data is represented with a box
  - The ends of the box are at the first and third quartiles, i.e., the height of the box is IRQ
  - The median is marked by a line within the box
  - Whiskers: two lines outside the box extend to Minimum and Maximum

Histogram Analysis

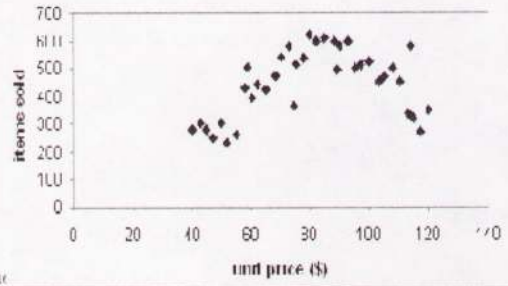
continuous attributes.

- Graph displays of basic statistical class descriptions
  - Frequency histograms
    - A univariate graphical method
    - Consists of a set of rectangles that reflect the counts or frequencies of the classes present in the given data



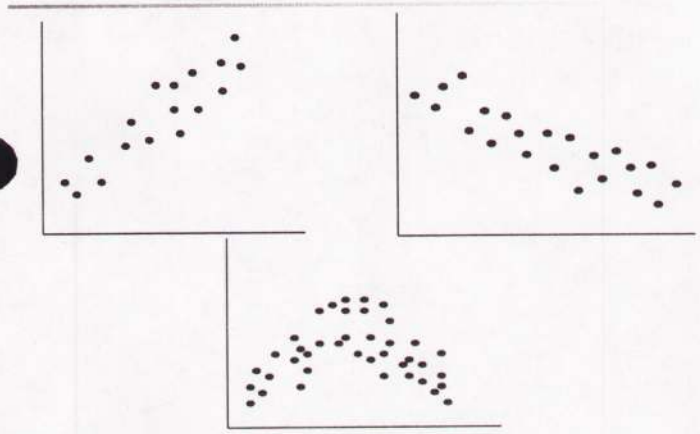
Scatter plot

- Provides a first look at bivariate data to see clusters of points, outliers, etc
- Each pair of values is treated as a pair of coordinates and plotted as points in the plane

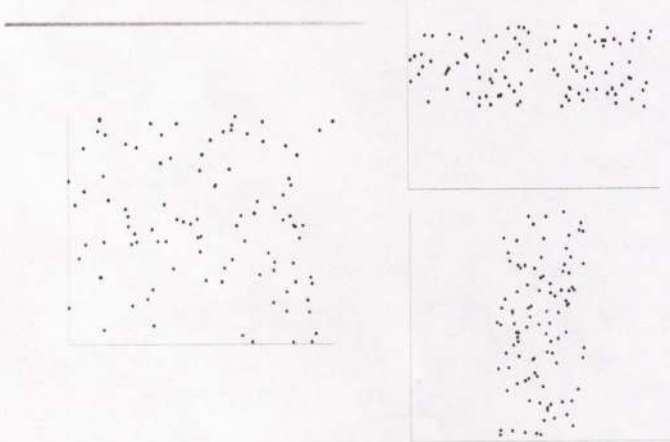


distribution for discrete attributes.

Positively and Negatively Correlated Data



Not Correlated Data



Modeler Demo of Visualization

- Graphing Nodes → discrete.
  - Histogram/Distribution Node → continuous.
  - Plot Node
  - Collection Node
  - Multiplot Node
  - Graphboard
    - This node has many different graphing options

References

- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Communications of ACM, 42:73-78, 1999
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley & Sons, 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), December 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
- Y. Wand and R. Wang. Anchoring data quality dimensions ontological foundations. Communications of ACM, 39:86-95, 1996
- 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

Lecture 3. Data Cleaning and Integration

Review from Last Lecture and Today's Content

- Last time we saw:
- Basic stream building overview
  - Basic nodes like type, filter, derive, etc.
- Data Cleaning techniques in Modeler
  - Data Audit Node
- Graphs for Visualization
  - Graphboard and other Graph Nodes
- Today we will continue with data cleaning, integration, transformation, and discretization

1

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

2

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?

3

Why Is Data Preprocessing Important?

- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
    - e.g., duplicate or missing data may cause incorrect or even misleading statistics.
  - Data warehouse needs consistent integration of quality data
- Data extraction, cleaning, and transformation comprises the majority of the work of building a data warehouse

Major Tasks in Data Preprocessing

1. **Data cleaning**
   - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
2. **Data integration**
   - Integration of multiple databases, data cubes, or files
3. **Data reduction**
   - Dimensionality reduction
   - Numerosity reduction
   - Data compression
4. **Data transformation and data discretization**
   - Normalization
   - Concept hierarchy generation

5

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

6

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?

7

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

8

How to Handle Missing Data?

1. Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
2. Fill in the missing value manually: tedious + infeasible?
3. 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

9

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

10

We can do PCA without missing values, if result shows it's not principle components, we can still use those items.

How to Handle Noisy Data?

1. 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.
2. Regression
   - smooth by fitting the data into regression functions
3. Clustering
   - detect and remove outliers
4. Combined computer and human inspection
   - detect suspicious values and check by human (e.g., deal with possible outliers)

11

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)

12

16

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

Data Integration

- Data integration:
  - Combines data from multiple sources into a coherent store
- Schema integration: e.g., A.cust-id ≡ 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

Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases – **Merge Node – will see examples later**
  - *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
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality – some methods handle correlated variables better than others

Correlation Analysis (Numerical Data)

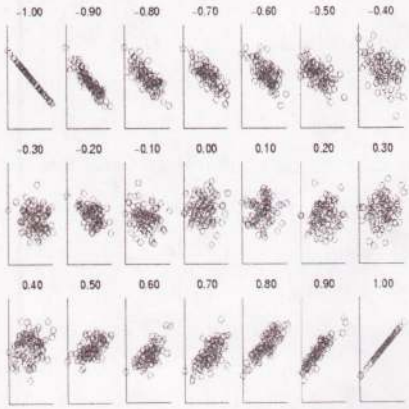
- Correlation coefficient (also called Pearson's product moment coefficient)

$$r_{A,B} = \frac{\sum(A-\bar{A})(B-\bar{B})}{(n-1)\sigma_A\sigma_B} = \frac{\sum(AB)-n\bar{A}\bar{B}}{(n-1)\sigma_A\sigma_B}$$

where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B, and Σ(AB) is the sum of the AB cross-product.

- If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.
- $r_{A,B} = 0$: independent; $r_{A,B} < 0$: negatively correlated
- **Statistics and Feature Selection Nodes in Modeler**

Visually Evaluating Correlation



Scatter plots showing the similarity from –1 to 1.

Covariance (Numeric Data)

- Covariance is similar to correlation

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

Correlation coefficient: $r_{A,B} = \frac{Cov(A,B)}{\sigma_A\sigma_B}$

where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective mean or **expected values** of A and B, σ_A and σ_B are the respective standard deviation of A and B.

- **Positive covariance**: If $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values.
- **Negative covariance**: If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value.
- **Independence**: $Cov_{A,B} = 0$ but the converse is not true:
  - Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence

Correlation Analysis (Categorical Data)

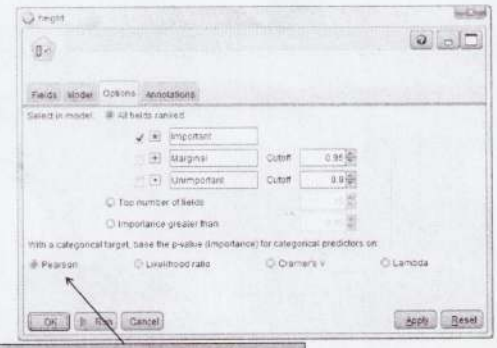
- X^2 (chi-square) test

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

- The larger the X^2 value, the more likely the variables are related
- The cells that contribute the most to the X^2 value are those whose actual count is very different from the expected count
- **Matrix Node and Feature Selection Nodes**
- **Correlation does not imply causality**
  - # of hospitals and # of car-theft in a city are correlated
  - Both are causally linked to the third variable: population

19

This is the Feature Selection Node Option Tab



Pearson is the Chi-Square Test

20

Chi-Square Calculation: An Example (See Slide notes for expected counts help)

	Play chess	Not play chess	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

- **Need at least 5 expected values in each cell (preferably more)** for chi-square test
- X^2 (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- Consult chi-square table with (rows-1)*(cols-1) = 1 degrees freedom. It is 10.28 at significance 0.001 and thus shows that like_science_fiction and play_chess are correlated in the group

21

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

22

Data Reduction Strategies

- Why data reduction?
  - A database/data warehouse may store terabytes of data
  - Complex data analysis/mining may take a very long time to run on the complete data set
- Data reduction
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results
- Data reduction strategies – will cover regression, cluster analysis, CHAID (like Chimerge), C5.0 (entropy) later
  - Data cube aggregation:
  - Dimensionality reduction — e.g., remove unimportant attributes
  - Data Compression
  - Numerosity reduction — e.g., fit data into models
  - Discretization and concept hierarchy generation

23

Data Cube Aggregation

- **Aggregate Node in Modeler**
- The lowest level of a data cube (base cuboid)
  - The aggregated data for an individual entity of interest
  - E.g., a customer in a phone calling data warehouse
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

24

Attribute Subset Selection

- **Feature Selection Node** (i.e., attribute subset selection):
  - Select a minimum set of features such that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features
  - reduce # of patterns in the patterns, easier to understand
- Heuristic methods – **Options in some Modeler Nodes (e.g. regression nodes)** (due to exponential # of choices):
  - Step-wise forward selection
  - Step-wise backward elimination
  - Combining forward selection and backward elimination

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

Data Transformation

- Smoothing: remove noise from data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Attribute/feature construction
  - New attributes constructed from the given ones

Data Transformation: Normalization – Auto Data Preparation Node

1) 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 $\frac{73,600-12,000}{98,000-12,000}(1.0-0)+0=0.716$

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

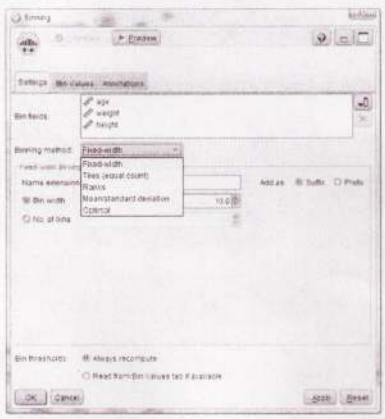
$$v' = \frac{v - \mu_A}{\sigma_A}$$

- Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73,600-54,000}{16,000}=1.225$
- **Modeler Auto Data Prep Node can do this automatically**

Simple Discretization Methods: Binning Node

- 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

Binning Node pull-down allows selection of binning type



References

- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Communications of ACM, 42:73-78, 1999
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley & Sons, 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), December 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
- Y. Wand and R. Wang. Anchoring data quality dimensions ontological foundations. Communications of ACM, 39:86-95, 1996
- 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

Lecture 4. Modeling and Linear Prediction.

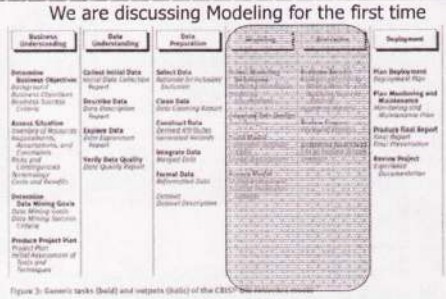
Linear Prediction Modeling and Model Assessment

1

(CRoss-Industry Standard Process for Data Mining)

CRISP-DM Data Mining Model

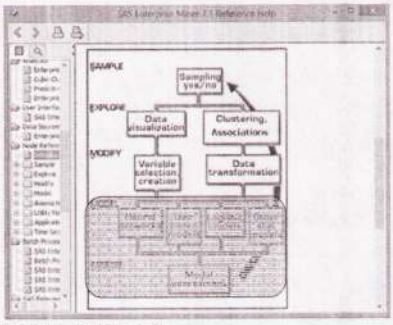
We are discussing Modeling for the first time



Source: CRISP-DM 1.0 Guide from SPSS.com. Also see www.crisp-dm.org

2

SEMMA Process



Source: SAS Enterprise Miner help

3

Evaluating the Accuracy of a Classifier or Predictor (I) – Dataset Partitioning

(pp. 370-371 in textbook)

- Holdout method – Sample Tab **Data Partition Node** in EMiner
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation **(Note EM calls this set the validation set)**
  - Random sampling: a variation of holdout
    - Repeat holdout k times, accuracy = avg. of the accuracies obtained
- Cross-validation (*k*-fold, where k = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set
  - Leave-one-out: k folds where k = # of tuples, for small sized data
  - Stratified cross-validation: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data

training set / test set

4

Evaluating the Accuracy of a Classifier or Predictor (II) - Dataset Partitioning

- Bootstrap
  - Works well with small data sets
  - Samples the given training tuples uniformly *with replacement*
    - i.e., each time a tuple is selected, it is equally likely to be selected again and re-added to the training set
- Several boostrap methods, and a common one is **.632 boostrap**
  - Suppose we are given a data set of d tuples. The data set is sampled d times, with replacement, resulting in a training set of d samples. The data tuples that did not make it into the training set end up forming the test set. About 63.2% of the original data will end up in the bootstrap, and the remaining 36.8% will form the test set (since $(1 - 1/d)^d \approx e^{-1} = 0.368$)
  - Repeat the sampling procedue k times, overall accuracy of the model:

$$acc(M) = \sum_{i=1}^{k} (0.632 \times acc(M_i)_{test\_set} + 0.368 \times acc(M_i)_{train\_set})$$

5

Classification vs. Prediction

difference

- Classification - Will see soon
  - predicts categorical class labels (discrete or nominal)
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Prediction
  - models continuous-valued functions, i.e., predicts unknown or missing values
- Typical applications
  - Credit approval
  - Target marketing
  - Medical diagnosis
  - Fraud detection

Classification vs. Prediction.

What Is (Numerical) Prediction?

- (Numerical) prediction is similar to classification
  - construct a model
  - use model to predict continuous or ordered value for a given input
- Prediction is different from classification
  - Classification refers to predict categorical class label
  - Prediction models continuous-valued functions
- Major method for prediction: regression
  - model the relationship between one or more *independent* or **predictor** (X) variables and a *dependent* or **response** (Y) variable
- Regression analysis
  - Linear and multiple regression
  - Non-linear regression
  - Other regression methods: generalized linear model, Poisson regression, log-linear models, regression trees

Linear Regression Review

- Linear regression: involves a response variable y and a single predictor variable x

  $y = w_0 + w_1 x$

  where w_0 (y-intercept) and w_1 (slope) are regression coefficients
- Method of least squares: estimates the best-fitting straight line

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2} \qquad w_0 = \bar{y} - w_1 \bar{x}$$

- Multiple linear regression: involves more than one predictor variable
  - Training data is of the form $(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_{|D|}, y_{|D|})$
  - Ex. For 2-D data, we may have: $y = w_0 + w_1 x_1 + w_2 x_2$
  - Solvable by extension of least square method or using Modeler
  - Many nonlinear functions can be transformed into the above

Linear Regression Example

Assumptions

- Assumptions for Linear Regression
- **(i) linearity** of the relationship between dependent and independent variables
- **(ii) independence** of the errors (no serial correlation)
- **(iii) homoscedasticity** (constant variance) of the errors
- (a) versus time
- (b) versus the predictions (or versus any independent variable)
- **(iv) normality** of the error distribution.

• Source: http://www.duke.edu/~rnau/testing.htm
• http://www.statsoft.com/textbook/multiple-regression/#assumptions also has a nice write-up on this

Multicollinearity in Multiple Linear Regression

- See http://www.graphpad.com/support/faqid/1768/
- Multicollinearity occurs when two or more predictor variables are correlated to each other
- This does not affect predictions, but does affect interpretation of predictor variables (x's) affect on target variable (y)
  - Important variables may not be significant
  - Confidence intervals may be very wide
- We will use correlation tests, graphs, and Regression Node to detect and eliminate multicollinearity

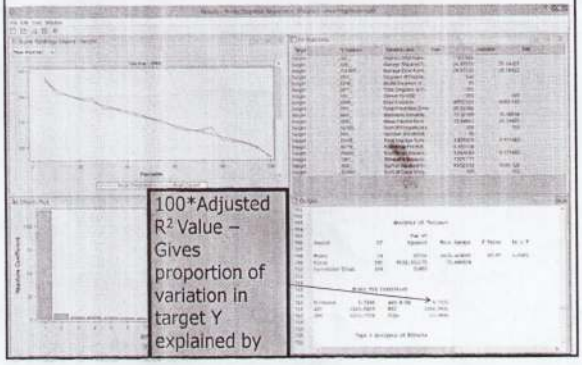
3 Types of Linear Node Model Selection

- **Selection Model** — Use the Selection Model property of the Regression node to specify the model selection method that you want to use during training. You can choose from the following effect selection methods:
  1. **Backward** — begins with all candidate effects in the model and removes effects until the Stay Significance Level or the Stop Criterion is met.
  2. **Forward** — begins with no candidate effects in the model and adds effects until the Entry Significance Level or the Stop Criterion is met.
  3. **Stepwise** — begins as in the forward model but may remove effects already in the model. Continues until Stay Significance Level or Stepwise Stopping Criteria are met.
  4. **None** — (default setting) all inputs are used to fit the model.

Source: EMiner Help

may not provide optimal set

Regression Node Model Results for Linear Regression



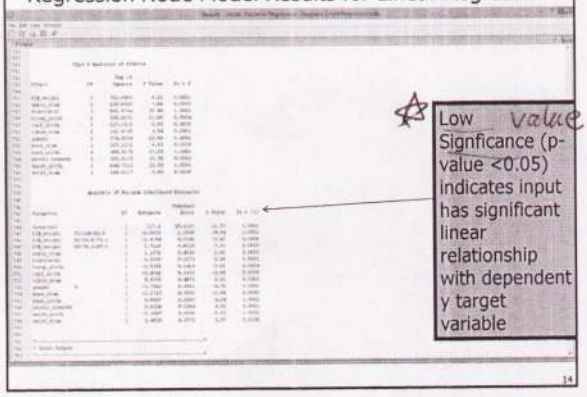
100*Adjusted R^2 Value – Gives proportion of variation in target Y explained by

R^2 vs. Adjusted R^2

- Adjusted R^2 takes the degrees of freedom of the error and total into account.
- In practice, R^2 will only increase or stay the same when adding variables to the model
- Adjusted R^2 accounts for this so that adding variables to the model can decrease Adjusted R^2
  - This prevents overfitting of models with too many X variables to explain the Y
  - Usually for problems we will see, R^2 and Adjusted R^2 will be close

13

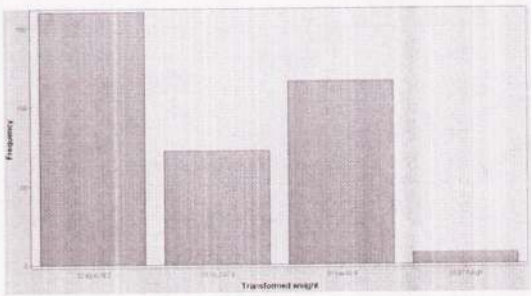
Regression Node Model Results for Linear Regression



Low Signficance (p-value <0.05) indicates input has significant linear relationship with dependent y target variable

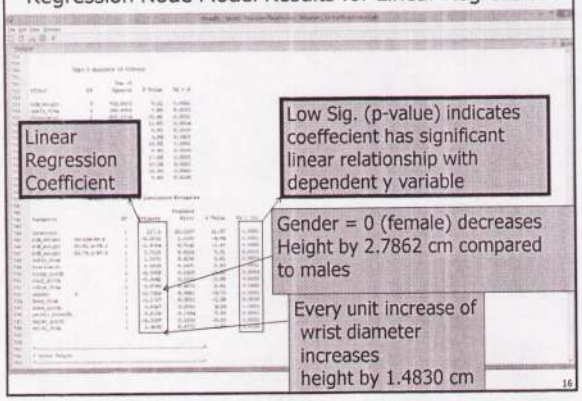
14

Weight Bins for this regression example where the target is height



15

Regression Node Model Results for Linear Regression



Linear Regression Coefficient

Low Sig. (p-value) indicates coeffecient has significant linear relationship with dependent y variable

Gender = 0 (female) decreases Height by 2.7862 cm compared to males

Every unit increase of wrist diameter increases height by 1.4830 cm

16

Predictor Error Measures – **Regression Node Results Window**

- Measure predictor accuracy: measure how far off the predicted value is from the actual known value
- **Loss function**: measures the error betw. y_i and the predicted value y_i'
  - Absolute error: $| y_i - y_i' |$
  - Squared error: $(y_i - y_i')^2$
- Test error (generalization error): the average loss over the test set
  - Mean absolute error: $\frac{\sum_{i=1}^{d} |y_i - y_i'|}{d}$ Mean squared error: $\frac{\sum_{i=1}^{d} (y_i - y_i')^2}{d}$
  - Relative absolute error: $\frac{\sum_{i=1}^{d} |y_i - y_i'|}{\sum_{i=1}^{d} |y_i - \bar{y}|}$ Relative squared error: $\frac{\sum_{i=1}^{d} (y_i - y_i')^2}{\sum_{i=1}^{d} (y_i - \bar{y})^2}$

The mean squared-error exaggerates the presence of outliers

Popularly use (square) root mean-square error, similarly, root relative squared error

17

Regression Node Results Fit Statistics Window

There are many statistics to evaluate Linear Regression model performance. For our class, we will use Root Mean Square Error unless otherwise noted.

If the error is much higher on the validation data compared to the training data, this is an indication your model has overfit the training data

Lecture 5. PCA and Classification Model Evaluation

Class Agenda

- Principal Components Analysis Example using Linear Regression Example from Last Class' Lab
- Model Evaluation
  - We saw Prediction Evaluation previously
  - Today we will see Classification Model Evaluation
    - Model Comparison Node for Classification Model Evaluation

1

Dimensionality Reduction: Principal Component Analysis Node (PCA) (pp. 102-103 in text book)

- **Principal Components Node in EMiner**
- Given N data vectors from n-dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
- Steps
  - 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, it is possible to reconstruct a good approximation of the original data
- Works for numeric data only
- Used when the number of dimensions is large

2

Principal Components Node Example

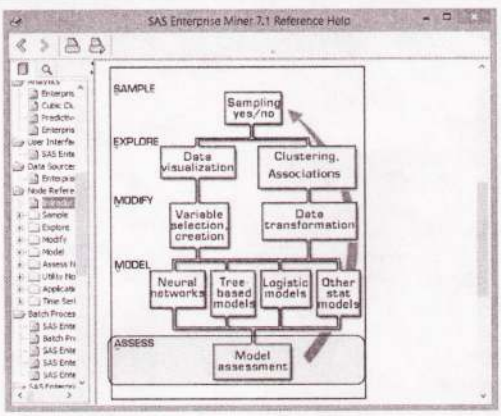
- We have a high-level definition of PCA
- Now, we will see how to run a PCA in Enterprise Miner to handle the multicollinearity present in the predictors of height from our linear regression lab.
- In PCA, the factors are ordered in the order in which they explain more dependent (target) variable variance
  - It may be possible to drop some of the less important factors as they may not add much to model quality – Use model evaluation techniques we will discuss to decide this

3

MODEL EVALUATION – SECTION 8.5 IN BOOK

4

SEMMA Process



Source: SAS Enterprise Miner help

5

Classification vs. Prediction

- Classification
  - predicts categorical class labels (discrete or nominal)
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Prediction
  - models continuous-valued functions, i.e., predicts unknown or missing values
- Typical applications
  - Credit approval
  - Target marketing
  - Medical diagnosis
  - Fraud detection

6

Classifier Accuracy Measures – **Model Comparison Node**

	C_1	C_2
C_1	True positive	False negative
C_2	False positive	True negative

classes	buy_computer = yes	buy_computer = no	total	recognition(%)
buy_computer = yes	6954	46	7000	99.34
buy_computer = no	412	2588	3000	86.27
total	7366	2634	10000	95.42

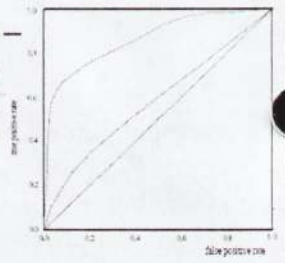
- Accuracy of a classifier M, acc(M): percentage of test set tuples that are correctly classified by the model M
  - Error rate (misclassification rate) of M = 1 – acc(M)
  - Given m classes, $CM_{i,j}$, an entry in a **coincidence (confusion in the book) matrix**, indicates # of tuples in class i that are labeled by the classifier as class j
- Alternative accuracy measures (e.g., for cancer diagnosis)

```
sensitivity = t-pos/pos          /* true positive recognition rate */
specificity = t-neg/neg          /* true negative recognition rate */
precision =  t-pos/(t-pos + f-pos)
accuracy = sensitivity * pos/(pos + neg) + specificity * neg/(pos + neg)
```

  - This model can also be used for cost-benefit analysis

7

Model Selection: ROC Curves – **Model Comparison Node**



- ROC (Receiver Operating Characteristics) curves: for visual comparison of classification models
- Originated from signal detection theory
- Shows the trade-off between the true positive rate and the false positive rate
- The area under the ROC curve is a measure of the accuracy of the model
- Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model
- Vertical axis represents the true positive rate
- Horizontal axis rep. the false positive rate
- The plot also shows a diagonal line
- A model with perfect accuracy will have an area of 1.0

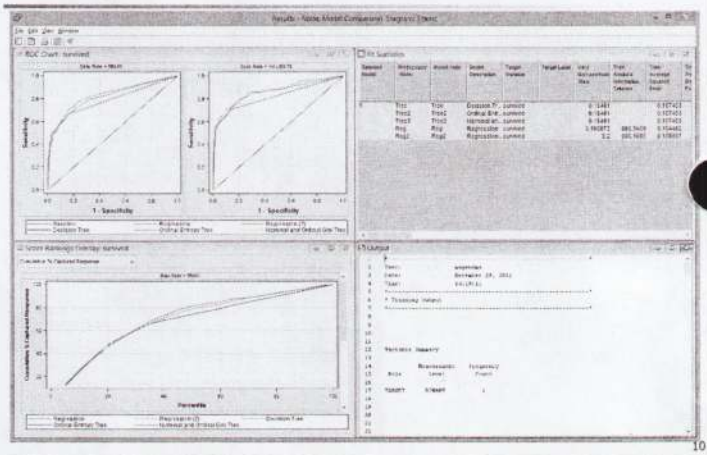
8

Model Comparison Node

- The Model Comparison node belongs to the Assess category in the SAS data mining process of Sample, Explore, Modify, Model, and Assess (SEMMA). The Model Comparison node enables you to compare the performance of competing models using various benchmarking criteria.
- There are many criteria that can be used to compare models. Comparing model performance depends on the specific application for the model. For example, with binary targets the Model Comparison node provides criteria that are derived from several sources.
- Classification Measures: comparative criteria include the Receiver Operating Characteristic (ROC) charts and corresponding area under the curve, classification rates, and so on.
- Data Mining Measures: comparative criteria from the study of data mining include lift and gain measures and profit and loss measures.
- Statistical Measures: comparative criteria from statistical literature include Bayesian Information Criterion (BIC), Akaike's Information Criterion (AIC), Gini statistics, Kolmogorov-Smirnov statistics, and Bin-Best Two-Way Kolmogorov-Smirnov tests.
- The combined criteria enable the analyst to make cross-model comparisons and assessments.
- When you train a modeling node, assessment statistics are computed on the train (and validation) data sets. The Model Comparison node computes the same statistics for the test set when it is present. **SOURCE: Eminer Help**

9

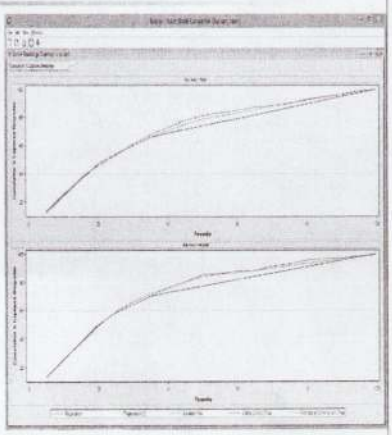
Model Comparision Node Results



10

Model Selection – Cumulative % Captured Chart **(Model Comparison Node)**

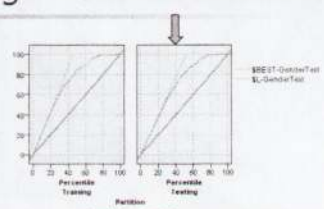
- Hit % Captured are defined as the proportion of total hits that occurs in each quantile. Gains are computed as (number of hits in quantile / total number of hits) × 100%.



11

Cumulative % Captured Chart Interpretation (Image from IBM Modeler)

- **Best line** is the light blue line. This line represents the best gain possible if you ordered all hits before any misses.
- Baseline is the red line. This line represents selecting at random. In other words, if you randomly selected 30% of the data, you would expect to get 30% of the hits.
- The brown line represents the model. It is better if this line is as close to the best line as possible
- Rank the test tuples in decreasing order

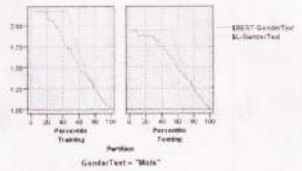


Interpretation of the Testing Set Cumulative % Captured Chart @ 40% sample: By using the model, we can identify 69.535% of the males. The best we can do is 77.519% males and the baseline is 40% males.

12

③ Cumulative Lift Chart (Image from IBM Modeler)

- This is the cumulative lift chart corresponding to the Cumulative % Captured Chart on previous slide.
- The best lift is the light blue line, the baseline is the red, and the model the brown line.
- Baseline lift is defined as 1.
- The best lift for the Testing Set is 1.938. This is because there are 51.6% males in the testing set and 1/.516 = 1.938
- Lift can be derived from Cumulative % Captured
- Lift is the multiple of improvement of the model hit rate over the baseline hit rate.



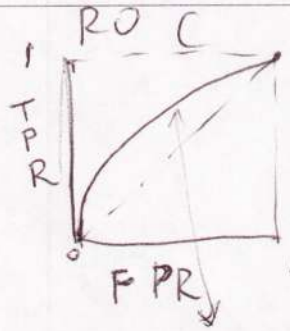
In our previous gains chart, at 40% sample we saw a 69.5% male hit rate with the model. This means the lift of the model over the baseline is 69.535/40 = 1.738

13

In this lecture, we saw quantitative measures for evaluating models – what role do qualitative issues play?

- Things to consider
  - Simple Models are Better Models all else being equal
  - People Issues
    - Model complexity may make explaining model difficult (e.g. Principal Components)
    - Are there people on the project who have pre-determined opinions? May need to take this into account in determining which model is best
  - Ethical Issues
  - Other qualitative issues...

14



$TPR = \frac{TP}{P}$

$FPR = \frac{FP}{N} = 1 - \frac{TN}{N}$

its similar with % of population with descending propensity.

		Predicted class		
		1	0	
Actual class	1	TP	FN	P
	0	FP	TN	N.

It's a plot of the ~~TPR~~ TPR v.s. FPR for the different possible cutpoints of a diagnostic test.

demonstrate: 1). the tradeoff between sensitivity ↑ and specificity ↓. P8.

2). For example:

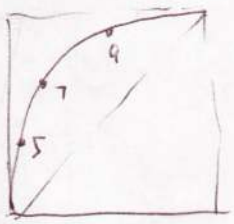
P : Hypothyroid : H.

N : Euthyroid : E.

T4 value	≤5	>5
H	18	14
E	1	92.

Cutpoint	Sensitivity	Sepcificity	FPR
5	0.56	0.99	0.01
7	0.78	0.81	0.19
9	0.91	0.42	0.58

T4 value	≤7	>7
H	25	7
E	18	75



The Area Under an ROC curve.

(0.9, 1) = excellent (A).

(0.8, 0.9) = good (B).

(0.7, 0.8) = fair (C).

(0.6, 0.7) = poor (D).

(0.5, 0.6) = fail (F)

Ty value	≤9	>9
H	29	3
E	54	39

Propensity = how much the model believes its prediction is "True" (positive)

If propensity ≥ 0.5, then prediction is "~~true~~". Positive

Confidence = how much the model believes its prediction is correct.

$\in [0.5, 1]$.

If prediction = "True". Propensity = confidence.

If prediction = "False". confidence = 1 − propensity.

Example:

(descending)

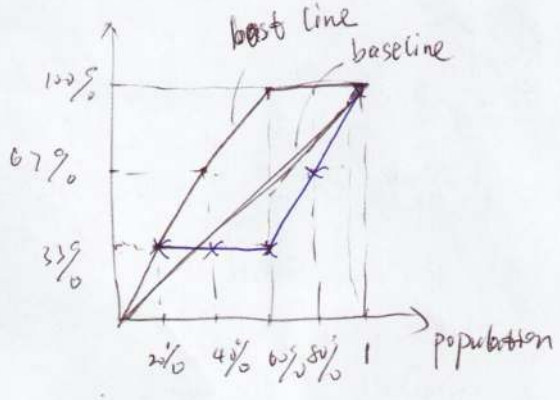
Actual	prediction	Propensity	confidence
T	T	1	1
F	T	0.9	0.9
F	T	0.75	0.75
T	T	0.6	0.6
T	F	0.45	0.55

It's about the rank of propensity.

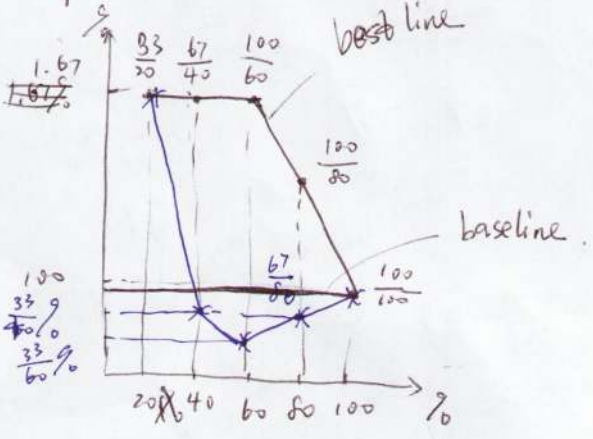
a good line means a good ranking predictor.

Based on this, we can change the threshold. (propensity > 0.5 to like pr > 0.3) to get a good classifier.

Gain chart (Cumulative % Captured Chart)



Lift chart.



Lecture 6. Decision Trees.

Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- Decision Tree Induction
- Bayes Classification Methods
- Rule-Based Classification
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary

1

Supervised vs. Unsupervised Learning

- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

2

Prediction Problems: Classification vs. Numeric Prediction

- Classification
  - predicts categorical class labels (discrete or nominal)
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Numeric Prediction
  - models continuous-valued functions, i.e., predicts unknown or missing values
- Typical applications
  - Credit/loan approval:
  - Medical diagnosis: if a tumor is cancerous or benign
  - Fraud detection: if a transaction is fraudulent
  - Web page categorization: which category it is

3

Classification—A Two-Step Process

step 1

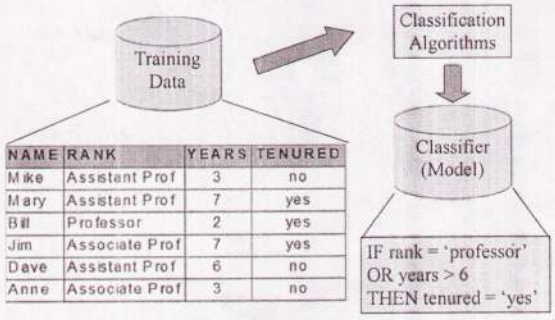
- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction is training set
  - The model is represented as classification rules, decision trees, or mathematical formulae

step 2

- Model usage: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model
    - Test set is independent of training set (otherwise overfitting)
  - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

4

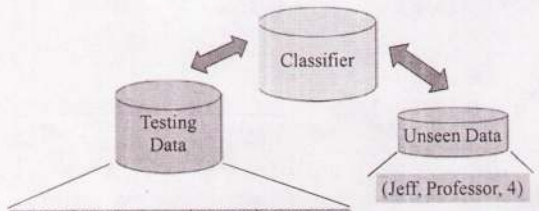
Process (1): Model Construction



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

5

Process (2): Using the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

6

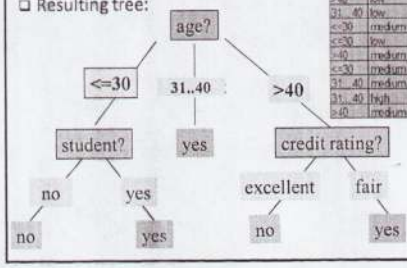
Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- Decision Tree Induction
- Bayes Classification Methods
- Rule-Based Classification
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary

7

Decision Tree Induction: An Example

- Training data set: Buys_computer
- The data set follows an example of Quinlan's ID3 (Playing Tennis)
- Resulting tree:



age	income	student	credit rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

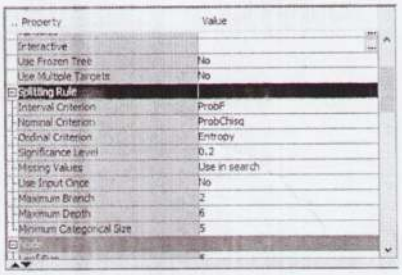
8

Eminer Decision Tree Node Train Properties: Splitting Rule

1. **Interval Criterion** — specify the method that you want to use to evaluate candidate splitting rules for interval variables and to search for the best one. Choose from the following splitting criteria:
   - **ProbF** — p-value of F-test associated with node variance.
   - **Variance** — reduction in the square error from node means.
2. **Nominal (and Binary) Criterion** — specify the method that you want to use to evaluate candidate splitting rules for nominal variables and to search for the best one. Choose from the following splitting criteria:
   - **ProbChisq** — p-value of Pearson Chi-square statistic for target vs. the branch node.
   - **Entropy** — reduction in entropy measure.
   - **Gini** — reduction in Gini index.
3. **Ordinal Criterion** — specify the method that you want to use to evaluate candidate splitting rules for ordinal variables and to search for the best one. Choose from the following splitting criteria:
   - **Entropy** — reduction in entropy measure, adjusted with ordinal distances.
   - **Gini** — reduction in Gini index, adjusted with ordinal distances.

9

Decision Tree Node Properties – Splitting Rule



10

EMiner Decision Tree Node Surrogate Splitting for Missing Input Values

- When a split is applied to an observation in which the required input value is missing, surrogate splitting rules can be considered before assigning the observation to the branch for missing values.
- A surrogate splitting rule is a back-up to the main splitting rule. For example, the main splitting rule might use COUNTY as input and the surrogate might use REGION. If the COUNTY is unknown and the REGION is known, the surrogate is used.
- If several surrogate rules exist, each surrogate is considered in sequence until one can be applied to the observation. If none can be applied, the main rule assigns the observation to the branch that is designated for missing values.
- SOURCE: EMiner Help

11

Summary of Common Decision Tree Methods (Can simulate with Eminer Decision Tree Node Settings)

Criterion	C5.0	CHAID	QUEST	C&R Tree (CART)
Split Type	Multiple	Multiple	Binary	Binary
Continuous Target?	No	Yes	No	Yes
Criterion for Predictor Selection	Gain Ratio	Chi-square/ F-test for Continuous	Statistical (Chi-square and F Test plus discriminant analysis)	Impurity (Gini for categorical/least squared for continuous
Missing Predictor Values?	Uses fractionalization	Yes, missing values are valid category	Uses Surrogates	Use Surrogates
Pruning	Upper limit on predicted error	Stops before overfit	Pruning	Pruning

12

Characteristics of a Good Decision Tree

- Pure nodes (Nodes with mostly one category of the target variable)
- Simple is better (Less levels and less nodes preferable to many levels/nodes)
- All nodes should have a significant number of cases (rule of thumb: at least 10 and hopefully more for most problems)

13

Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

3 ways

14

Introduction to Entropy for Uncertainty (impurity in decision trees)

- In Information Systems, entropy, known as Shannon entropy for Claude Shannon, is the measure of uncertainty in a random variable. A coin toss has one bit of entropy for the 50/50 chance of it turning up heads or tails, 0 or 1. A six-sided dice carries three bits of entropy for the possible outcomes it may produce with each roll (1 (000), 2 (001), 3 (010), 4 (011), 5 (100), 6 (101)). The weather has an amount of entropy difficult to quantify, but it varies from location to location. The weather in New York has more entropy than the weather in Southern California because Southern California has a more consistent climate. Similarly, in our first example, if we were dealing with a rigged coin, one that turned up heads more often than tails, then there would be less than one bit of entropy in each coin toss because we would expect heads more frequently than tails.
- From http://ideonexus.com/2010/08/30/entropy-for-information-systems/

15

Attribute Selection Measure: Information Gain (ID3/C4.5) – There is a newer C5.0 version

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

16

Attribute Selection: Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$\frac{5}{14} I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes and 3 no.

Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

17

Gain Ratio for Attribute Selection (C4.5)

- Information gain measure is biased towards attributes with a large number of values
- C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

- GainRatio(A) = Gain(A)/SplitInfo(A)
- Ex.

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

  - gain_ratio(income) = 0.029/1.557 = 0.019
- The attribute with the maximum gain ratio is selected as the splitting attribute

18

Gini index ⇒ binary partitions (split)

Gain Ratio ⇒ multiple partitions. (split)

Gain Ratio for Attribute Selection (C4.5)

- $Info_{age}(D) = \frac{5}{14}(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}) + \frac{4}{14}(-\frac{4}{4}\log_2\frac{4}{4}) + \frac{5}{14}(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}) = 0.694$
- $Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246$
- GainRatio(A) = Gain(A)/SplitInfo(A)
- Ex. $SplitInfo_A(D) = -\sum_{j=1}^{v}\frac{|D_j|}{|D|}\times\log_2(\frac{|D_j|}{|D|})$

  $SplitInfo_A(D) = -\frac{5}{14}\times\log_2(\frac{5}{14}) - \frac{4}{14}\times\log_2(\frac{4}{14}) - \frac{5}{14}\times\log_2(\frac{5}{14}) = 1.577$
  - gain_ratio(age) = 0.246/1.577 = 0.152
- The attribute with the maximum gain ratio is selected as the splitting attribute

19

Computing Information-Gain for Continuous-Valued Attributes

- Let attribute A be a continuous-valued attribute
- Must determine the *best split point* for A
  - Sort the value A in increasing order
  - Typically, the midpoint between each pair of adjacent values is considered as a possible *split point*
    - $(a_i+a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
  - The point with the *minimum expected information requirement* for A is selected as the split-point for A
- Split:
  - D1 is the set of tuples in D satisfying A ≤ split-point, and D2 is the set of tuples in D satisfying A > split-point

20

3 Gini Index (CART uses this)

- If a data set D contains examples from n classes, gini index, $gini(D)$ is defined as $gini(D) = 1 - \sum_{j=1}^{n} p_j^2$

  where p_j is the relative frequency of class j in D
- If a data set D is split on A into two subsets D_1 and D_2, the *gini* index $gini(D)$ is defined as $gini_A(D) = \frac{|D_1|}{|D|}gini(D_1) + \frac{|D_2|}{|D|}gini(D_2)$
- Reduction in Impurity: $\Delta gini(A) = gini(D) - gini_A(D)$
- The attribute that provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

21

Computation of Gini Index

- Ex. D has 9 tuples in buys_computer = "yes" and 5 in "no" $gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$
- Suppose the attribute income partitions D into 10 in D_1: {low, medium} and 4 in D_2 $gini_{income \in \{low,medium\}}(D) = \left(\frac{10}{14}\right)Gini(D_1) + \left(\frac{4}{14}\right)Gini(D_2)$

  $= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right)$

  $= 0.443$

  $= Gini_{income \in \{high\}}(D)$.

$Gini_{\{low,high\}}$ is 0.458; $Gini_{\{medium,high\}}$ is 0.450. Thus, split on the {low,medium} (and {high}) since it has the lowest Gini index

22

Comparing Attribute Selection Measures

- The three measures, in general, return good results but
  - **Information gain**:
    - biased towards multivalued attributes
  - **Gain ratio**:
    - tends to prefer unbalanced splits in which one partition is much smaller than the others
  - **Gini index**:
    - biased to multivalued attributes
    - has difficulty when # of classes is large
    - tends to favor tests that result in equal-sized partitions and purity in both partitions

23

CHAID - Chi-square Automatic Interaction Detection

- Performs Chi-Square test to split categorical variables
- Performs F Test to split continuous variables
- Treats Missing Values as valid values
- Sometimes gets "Split happy" and creates many nodes with few cases which is undesirable

24

Common Attribute Selection Measures

- CHAID: a popular decision tree algorithm, measure based on χ^2 test for independence
- C-SEP: performs better than info. gain and gini index in certain cases
- G-statistic: has a close approximation to χ^2 distribution
- MDL (Minimal Description Length) principle (i.e., the simplest solution is preferred):
  - The best tree as the one that requires the fewest # of bits to both (1) encode the tree, and (2) encode the exceptions to the tree
- Multivariate splits (partition based on multiple variable combinations)
  - CART: finds multivariate splits based on a linear comb. of attrs.
- Which attribute selection measure is the best?
  - Most give good results, none is significantly superior than others

25

Overfitting and Tree Pruning

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: *Halt tree construction early*-do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: *Remove branches* from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"

26

Decision Tree Subtree Selection (Tree Pruning) from Eminer Help

- **Assessment Measure** – Use the Assessment Measure property of the Decision Tree node to specify the method that you want to use to select the best tree, based on the validation data when the Method property is set to Assessment. If no validation data is available, training data is used. The available assessment measurements are:
  - **Decision** (default setting) — The Decision method selects the tree that has the largest average profit and smallest average loss if a profit or loss matrix is defined. If no profit or loss matrix is defined, the value of model assessment measure will be reset in the training process, depending on the measurement level of the target. If the target is interval, the measure is set to Average Square Error. If the target is categorical, the measure is set to Misclassification.
  - **Average Square Error** — The Average Square Error method selects the tree that has the smallest average square error.
  - **Misclassification** — The Misclassification method selects the tree that has the smallest misclassification rate.
  - **Lift** — The Lift method evaluates the tree based on the prediction of the top n% of the ranked observations. Observations are ranked based on their posterior probabilities or predicted target values. For an interval target, it is the average predicted target value of the top n% observations. For a categorical target, it is the proportion of events in the top n% data. When you set the Measure property to Lift, you must use the Assessment Fraction property to specify the proportion for the top n% of cases.

27

Advantage and Disadvantage of PCA.

A: Can remove correlations in original predictors by reducing the number of variables to new variables created by PCA from the original variables which are not correlated with each other.

D: PCA is hard to explain. PCA only works on numeric data.

Advantage and Disadvantage of Decision Tree with more tree depth.

A: More complex functions can be approximated

D: More difficult to explain, overfitting possible

Give 2 characteristics of a numeric input variable that would favor using ordinal measurement level instead of interval when building a decision tree.

1). The variable has relatively few values so that it can be treated as a categorical variable.

2). You want the tree to group values for the variable which are not contiguous (next to each other) in the same node. Interval measurement level cannot do this as it must group by continuous ranges.

The same and difference of Validation and Testing data.

Both the validation and testing sets are held-out sets to evaluate model results on data not used to build the model. However, the validation set is used iteratively as models as built to allow model comparison and changes to improve the model. The testing set is held out until the end of modeling to allow a final model evaluation of model contenders with data that hasn't been seen by the models or makers at all during the modeling process.

Lecture 7.

Data Mining:

Concepts and Techniques

(3rd ed.)

— Chapter 8.6 —

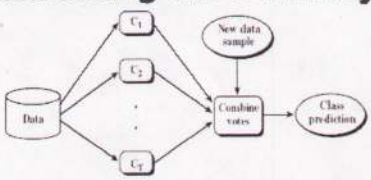
Jiawei Han, Micheline Kamber, and Jian Pei
University of Illinois at Urbana-Champaign &
Simon Fraser University

1

Chapter 8. Classification: Basic Concepts

2

Ensemble Methods: Increasing the Accuracy



- Ensemble methods
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, M_1, M_2, ..., M_k, with the aim of creating an improved model M*
- Popular ensemble methods
  - Bagging: averaging the prediction over a collection of classifiers
  - Boosting: weighted vote with a collection of classifiers
  - Ensemble: combining a set of heterogeneous classifiers

3

Bagging: Boostrap Aggregation

- Analogy: Diagnosis based on multiple doctors' majority vote
- Training
  - Given a set D of *d* tuples, at each iteration *i*, a training set D_i of *d* tuples is sampled with replacement from D (i.e., bootstrap)
  - A classifier model M_i is learned for each training set D_i
- Classification: classify an unknown sample **X**
  - Each classifier M_i returns its class prediction
  - The bagged classifier M* counts the votes and assigns the class with the most votes to **X**
- Prediction: can be applied to the prediction of continuous values by taking the average value of each prediction for a given test tuple
- Accuracy
  - Often significantly better than a single classifier derived from D
  - For noise data: not considerably worse, more robust
  - Proved improved accuracy in prediction

4

Boosting

- Analogy: Consult several doctors, based on a combination of weighted diagnoses—weight assigned based on the previous diagnosis accuracy
- How boosting works?
  - **Weights** are assigned to each training tuple
  - A series of k classifiers is iteratively learned
  - After a classifier M_i is learned, the weights are updated to allow the subsequent classifier, M_{i+1}, to **pay more attention to the training tuples that were misclassified** by M_i
  - The final **M* combines the votes** of each individual classifier, where the weight of each classifier's vote is a function of its accuracy
- Boosting algorithm can be extended for numeric prediction
- Comparing with bagging: Boosting tends to have greater accuracy, but it also risks overfitting the model to misclassified data

5

Adaboost (Freund and Schapire, 1997)

- Given a set of *d* class-labeled tuples, $(\mathbf{X}_1, y_1), \ldots, (\mathbf{X}_d, y_d)$
- Initially, all the weights of tuples are set the same (1/d)
- Generate k classifiers in k rounds. At round i,
  - Tuples from D are sampled (with replacement) to form a training set D_i of the same size
  - Each tuple's chance of being selected is based on its weight
  - A classification model M_i is derived from D_i
  - Its error rate is calculated using D_i as a test set
  - If a tuple is misclassified, its weight is increased, o.w. it is decreased
- Error rate: $err(\mathbf{X}_j)$ is the misclassification error of tuple $\mathbf{X}_j$. Classifier M_i error rate is the sum of the weights of the misclassified tuples:

$$error(M_i) = \sum_{j}^{d} w_j \times err(\mathbf{X}_j)$$

- The weight of classifier M_i's vote is $\log \frac{1 - error(M_i)}{error(M_i)}$

6

Random Forest (Breiman 2001)

- Random Forest:
  - Each classifier in the ensemble is a *decision tree* classifier and is generated using a random selection of attributes at each node to determine the split
  - During classification, each tree votes and the most popular class is returned
- Two Methods to construct Random Forest:
  - Forest-RI (*random input selection*): Randomly select, at each node, F attributes as candidates for the split at the node. The CART methodology is used to grow the trees to maximum size
  - Forest-RC (*random linear combinations*): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers)
- Comparable in accuracy to Adaboost, but more robust to errors and outliers
- Insensitive to the number of attributes selected for consideration at each split, and faster than bagging or boosting

7

Classification of Class-Imbalanced Data Sets

- Class-imbalance problem: Rare positive example but numerous negative ones, e.g., medical diagnosis, fraud, oil-spill, fault, etc.
- Traditional methods assume a balanced distribution of classes and equal error costs: not suitable for class-imbalanced data
- Typical methods for imbalance data in 2-class classification:
  - **Oversampling**: re-sampling of data from positive class
  - **Under-sampling**: randomly eliminate tuples from negative class
  - **Threshold-moving**: moves the decision threshold, t, so that the rare class tuples are easier to classify, and hence, less chance of costly false negative errors
  - Ensemble techniques: Ensemble multiple classifiers introduced above
- Still difficult for class imbalance problem on multiclass tasks

8

Lecture 8. NN

Section 9.2 Neural Networks

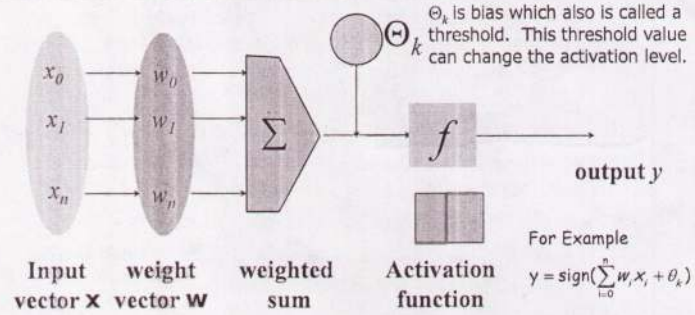
Classification by Backpropagation

- Backpropagation: A **neural network** learning algorithm
- Started by psychologists and neurobiologists to develop and test computational analogues of neurons
- A neural network: A set of connected input/output units where each connection has a **weight** associated with it
- During the learning phase, the **network learns by adjusting the weights** so as to be able to predict the correct class label of the input tuples
- Also referred to as **connectionist learning** due to the connections between units

Neural Network as a Classifier

- Weakness
  - Long training time
  - Require a number of parameters typically best determined empirically, e.g., the network topology or ``structure."
  - Poor interpretability: Difficult to interpret the symbolic meaning behind the learned weights and of ``hidden units" in the network
- Strength
  - High tolerance to noisy data
  - Ability to classify untrained patterns
  - Well-suited for continuous-valued inputs and outputs
  - Successful on a wide array of real-world data
  - Algorithms are inherently parallel
  - Techniques have recently been developed for the extraction of rules from trained neural networks

A Neuron (= a perceptron)

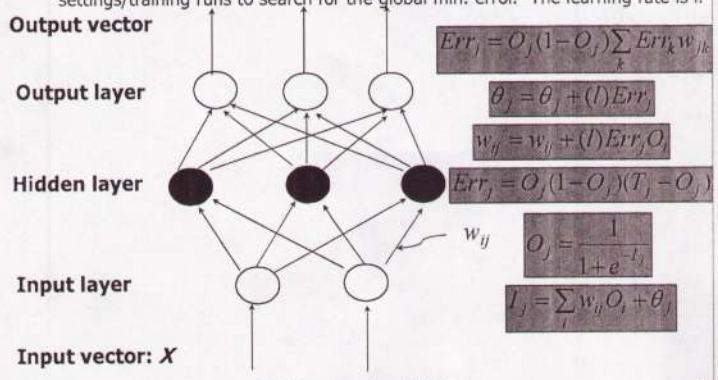


- The n-dimensional input vector **x** is mapped into variable y by means of the scalar product and a nonlinear function mapping

$g(x) = \frac{1}{1+e^{-y(x)}}$ $g'(x) = g(x)(1-g(x))$.

A Multi-Layer Feed-Forward Neural Network

$O_j(1-O_j)$ Is the derivative of the sigmoid (logistic) function. Backprop performs gradient descent on the error to find error **local minimum**. Try different initial weight settings/training runs to search for the global min. error. The learning rate is l.



How A Multi-Layer Neural Network Works?

- The **inputs** to the network correspond to the attributes measured for each training tuple 向量.
- Inputs are fed simultaneously into the units making up the **input layer**
- They are then weighted and fed simultaneously to a **hidden layer**
- The number of hidden layers is arbitrary, although usually only one
- The weighted outputs of the last hidden layer are input to units making up the **output layer**, which emits the network's prediction
- The network is **feed-forward** in that none of the weights cycles back to an input unit or to an output unit of a previous layer
- From a statistical point of view, networks perform **nonlinear regression**: Given enough hidden units and enough training samples, they can closely approximate any function

Defining a Network Topology

- First decide the **network topology:** # of units in the *input layer*, # of *hidden layers* (if > 1), # of units in *each hidden layer*, and # of units in the *output layer*
- Normalizing the input values for each attribute measured in the training tuples to [0.0—1.0]
- One **input** unit per domain value, each initialized to 0
- **Output**, if for classification and more than two classes, one output unit per class is used
- Once a network has been trained and its accuracy is **unacceptable**, repeat the training process with a *different network topology* or a *different set of initial weights*

Backpropagation

- Iteratively process a set of training tuples & compare the network's prediction with the actual known target value
- For each training tuple, the weights are modified to **minimize the mean squared error** between the network's prediction and the actual target value
- Modifications are made in the "**backwards**" direction: from the output layer, through each hidden layer down to the first hidden layer, hence "**backpropagation**"
- Steps
  - Initialize weights (to small random #s) and biases in the network
  - Propagate the inputs forward (by applying activation function)
  - Backpropagate the error (by updating weights and biases)
  - Terminating condition (when error is very small, etc.)

Backpropagation and Interpretability

- Efficiency of backpropagation: Each epoch (one interation through the training set) takes $O(|D| * w)$, with $|D|$ tuples and w weights, but # of epochs can be exponential to n, the number of inputs, in the worst case
- Rule extraction from networks: network pruning
  - Simplify the network structure by removing weighted links that have the least effect on the trained network
  - Then perform link, unit, or activation value clustering
  - The set of input and activation values are studied to derive rules describing the relationship between the input and hidden unit layers
- Sensitivity analysis: assess the impact that a given input variable has on a network output. The knowledge gained from this analysis can be represented in rules

Lecture #9: (& Association.

Chapter 6: Mining Frequent Patterns, Association and Correlations

- Basic concepts and a road map
- We will cover pages 243 – 254 and Sections 6.3.1 and 6.3.2 in the textbook.
- We will learn the Apriori algorithm in class, but Enterprise Miner has several options for association rules

What Is Frequent Pattern Analysis?

- Frequent pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of frequent itemsets and association rule mining
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?— Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?
- Applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.

Why Is Freq. Pattern Mining Important?

- Discloses an intrinsic and important property of data sets
- Forms the foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: associative classification
  - Cluster analysis: frequent pattern-based clustering
  - Data warehousing: iceberg cube and cube-gradient
  - Semantic data compression: fascicles
  - Broad applications

Frequent and Strong Rules; Antecedent and Consequent

- For a rule to be generated with Apriori, it must
  - Come from a frequent itemset
    - This means the minimum support is satisfied for the itemset
  - Be a strong rule
    - This means the rule must meet the minimum confidence (and minimum support since it comes from a frequent itemset)
- The form of a rule is "if antecedent then consequent"
  - For the rule A -> B, A is the antecedent and B is the consequent

Association Node Definitions from Enterprise Miner Help

$S = P(A \cup B)$

$C = \frac{P(A \cup B)}{P(A)}$

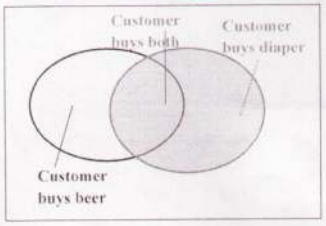
$E C = P(B)$

$L = \frac{P(A \cup B)}{P(A) P(B)}$

- Association discovery uses the following performance measures to evaluate association rules:
- *Support* — The level of support indicates how often the association combination occurs within the transaction database. In other words, support quantifies the probability of a transaction that contains both item A and item B.
- *Confidence* — The strength of an association is defined by its confidence factor. Given the association rule A ==> B, the confidence for the association rule is the conditional probability that a transaction contains item B, given that the transaction already contains item A.
- *Expected Confidence* — Given the association rule A ==> B, the expected confidence for the rule is the proportion of all transactions that contain item B. The difference between confidence and expected confidence is a measure of the change in predictive power due to the presence of item A in a transaction. Expected confidence provides an indication of what the confidence would be if there were no relationship between the items.
- *Lift* — Given the association rule A ==> B, the lift of the association rule is defined as the ratio of the rule's confidence to the rule's expected confidence. In other words, lift is the factor by which the confidence exceeds the expected confidence. Larger lift ratios tend to indicate more interesting association rules. The greater the lift, the greater the influence of an item A in a transaction has on the likelihood that item B will be contained in the transaction. Lift can be used as a general measure of the affinity that exists between the two items of interest.
- A creditable rule has a large confidence factor, a large level of support, and a value of lift greater than 1. Rules that have a high level of confidence, but have little support should be interpreted with caution.
- **See Enterprise Miner Help for formulas for the above definitions**

Basic Concepts: Frequent Patterns and Association Rules

Transaction-id	Items bought
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F



- Itemset X = $\{x_1, \ldots, x_k\}$
- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, s, probability that a transaction contains $X \cup Y$
  - confidence, c, conditional probability that a transaction having X also contains Y

Let $sup_{min} = 50\%$, $conf_{min} = 50\%$
Frequent. Pat.: {A:3, B:3, D:4, E:3, AD:3}
Association rules:
$A \rightarrow D$ (60%, 100%)
$D \rightarrow A$ (60%, 75%)

Closed Patterns and Max-Patterns

- A long pattern contains a combinatorial number of sub-patterns, e.g., $\{a_1, ..., a_{100}\}$ contains $\binom{100}{1} + \binom{100}{2} + ... + \binom{1}{1}\binom{0}{0}\binom{0}{0} = 2^{100} - 1 = 1.27*10^{30}$ sub-patterns!
- Solution: *Mine closed patterns and max-patterns instead*
- An itemset X is closed if X is *frequent* and there exists *no super-pattern* $Y \supset X$, *with the same support* as X (proposed by Pasquier, et al. @ ICDT'99)
- An itemset X is a max-pattern if X is frequent and there exists no frequent super-pattern $Y \supset X$ (proposed by Bayardo @ SIGMOD'98)
- Closed pattern is a lossless compression of freq. patterns
  - Reducing the # of patterns and rules

Closed Patterns and Max-Patterns

- Exercise 6.2 in book. DB = $\{<a_1, ..., a_{100}>, < a_1, ..., a_{50}>\}$
  - Min_sup = 1.
- What is the set of closed itemset?
  - $<a_1, ..., a_{100}>$: 1
  - $< a_1, ..., a_{50}>$: 2
- What is the set of max-pattern?
  - $<a_1, ..., a_{100}>$: 1
- Closed and max patterns allow efficient implementation of association rule algorithms

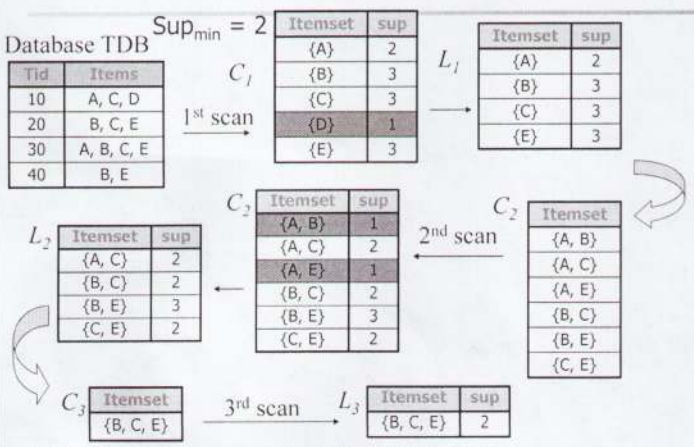
Scalable Methods for Mining Frequent Patterns

- The downward closure property of frequent patterns
  - Any subset of a frequent itemset must be frequent
  - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
  - i.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}
- Scalable mining methods: Three major approaches
  - Apriori (Agrawal & Srikant@VLDB'94)
  - Freq. pattern growth (FPgrowth—Han, Pei & Yin @SIGMOD'00)
  - Vertical data format approach (Charm—Zaki & Hsiao @SDM'02)

Apriori: A Candidate Generation-and-Test Approach

- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/tested! (Agrawal & Srikant @VLDB'94, Mannila, et al. @ KDD' 94)
- Method:
  - Initially, scan DB once to get frequent 1-itemset
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - Test the candidates against DB
  - Terminate when no frequent or candidate set can be generated

The Apriori Algorithm—An Example



The Apriori Algorithm

- Pseudo-code:

  C_k: Candidate itemset of size k
  L_k : frequent itemset of size k

  L_1 = {frequent items};
  for (k = 1; L_k !=∅; k++) **do begin**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
  increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
  end
  return $\cup_k L_k$;

Important Details of Apriori

- How to generate candidates?
  - Step 1: self-joining L_k
  - Step 2: pruning
- How to count supports of candidates?
- Example of Candidate-generation
  - L_3={*abc, abd, acd, ace, bcd*}
  - Self-joining: L_3*L_3
    - *abcd* from *abc* and *abd*
    - *acde* from *acd* and *ace*
  - Pruning:
    - *acde* is removed because *ade* is not in L_3
  - C_4={*abcd*}

Interestingness Measure: Correlations (Lift)

- *play basketball* ⇒ *eat cereal* [40%, 66.7%] is misleading
  - The overall % of students eating cereal is 75% > 66.7%.
- *play basketball* ⇒ *not eat cereal* [20%, 33.3%] is more accurate, although with lower support and confidence
- Measure of dependent/correlated events: lift >1 is good

$$lift = \frac{P(A \cup B)}{P(A)P(B)}$$

	Basketball	Not basketball	Sum (row)
Cereal	2000	1750	3750
Not cereal	1000	250	1250
Sum(col.)	3000	2000	5000

$$lift(B,C) = \frac{2000/5000}{3000/5000 * 3750/5000} = 0.89$$

$$lift(B,\neg C) = \frac{1000/5000}{3000/5000 * 1250/5000} = 1.33$$

Challenges of Frequent Pattern Mining

- Challenges
  - Multiple scans of transaction database
  - Huge number of candidates
  - Tedious workload of support counting for candidates
- Improving Apriori: general ideas
  - Reduce passes of transaction database scans
  - Shrink number of candidates
  - Facilitate support counting of candidates

Ref: Basic Concepts of Frequent Pattern Mining

- (Association Rules) R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93.
- (Max-pattern) R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98.
- (Closed-pattern) N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. ICDT'99.
- (Sequential pattern) R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95

Ref: Apriori and Its Improvements

- R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB'94.
- H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. KDD'94.
- A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. VLDB'95.
- J. S. Park, M. S. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD'95.
- H. Toivonen. Sampling large databases for association rules. VLDB'96.
- S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket analysis. SIGMOD'97.
- S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98.