

Knowledge Discovery in Very Large Databases



Xindong Wu
Department of Computer Science
University of Vermont
USA

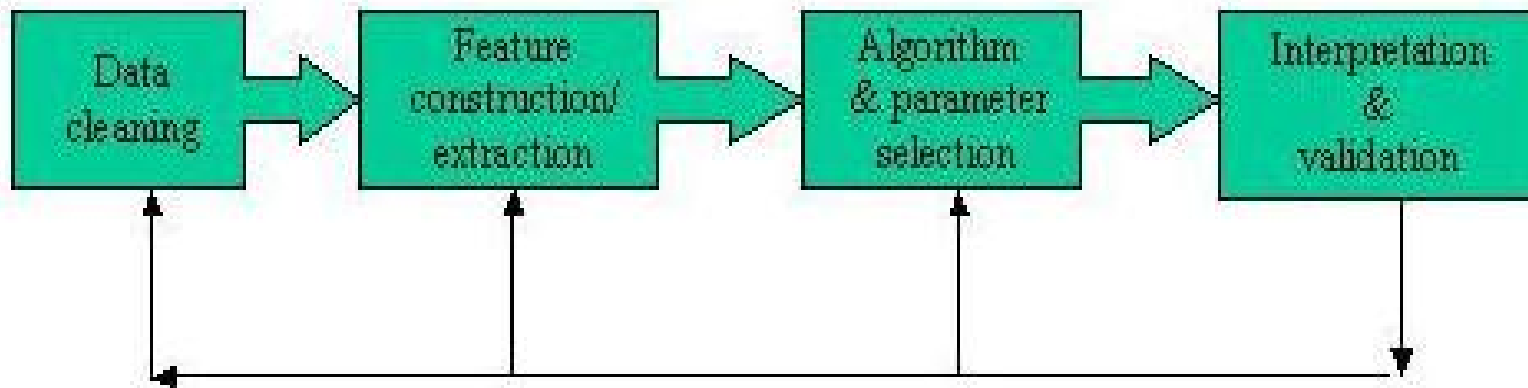
xwu@emba.uvm.edu
www.cs.uvm.edu/~xwu

Outline

- [Knowledge Discovery in Databases: An Overview](#)
- Data Mining at Large: Knowledge Engineering Support
 - Intelligent Learning Database Systems
- Dealing with Large Databases: Existing Data Mining Techniques
- Aggregating Rules from Different Data Sources
- Concluding Remarks

Knowledge Discovery in Databases (or Data Mining): An Introduction

- Definition: “identifying *valid, novel, potentially useful, and ultimately understandable* patterns in data” (Fayyad et al. 1996)
 - IEEE ICDM 2001: 365 submissions; ICDM 2002: 369.
- According to Philip Yu (Wu, Yu, et al. 2002):



Data Mining: Techniques and Applications

(with an insurance context)

- **Classification** of existing products, to determine the most appropriate policy for a new customer
- **Association analysis** to prevent customer loss (based on cancellations and unemployment for specific regions)
- **Cluster analysis:** Marketing to identify potential clients and to better serve policyholders.
- **Pattern recognition and deviation detection** in claims analysis for detecting areas of potential fraud
- **Simulation of changes and potential policies** (interpreting data mining results)

How to Select the Proper Techniques for Various Kinds of Projects

- Data characteristics
- Intended use of the mined knowledge
- Typical techniques
 - Classification
 - Rule induction
 - Decision tree construction
 - Association analysis
 - Statistical patterns
 - Neural networks
 - Genetic algorithms
 - Inductive logic programming

Problems with Data

- Data cleaning
 - Contradiction and redundancy elimination
 - Missing and stale data
- Data transforming – feature construction/extraction
 - Feature enhancement and dimension reduction
 - Mapping time-series data
- Data structuring and organization
 - Common terminology
 - Sampling strategy

Outline

- Knowledge Discovery in Databases: An Overview
- Data Mining at Large: Knowledge Engineering Support
 - Intelligent Learning Database Systems
- Dealing with Large Databases: Existing Data Mining Techniques
- Aggregating Rules from Different Data Sources
- Concluding Remarks

Mining at Large: Knowledge Management

- How to **handle massive bodies** of knowledge discovered from large, real-world data?
- When the data comes from different sources (e.g., from different hospitals), how to maintain the **consistency** of the knowledge that is extracted from each of these data sources?
- If the extracted knowledge **overfits** or **underfits** the data, how to resolve its incompleteness and contradiction?

Knowledge Management (2)

- No match: No knowledge extracted is applicable to a new problem.
- Multiple match: Knowledge from data mining gives conflicting advice to a new problem.
- Single match: a test example matches with the description of a specific class.

Intelligent Learning Database Systems

- An ILDB system (Wu 95, Wu 2000) provides mechanisms for
 - Preparation and translation of **database information** into a form suitable for use by its induction engines,
 - Using **induction** techniques to extract knowledge from databases, and
 - Interpreting the extracted knowledge to solve users' problems by **deduction** (or KBS technology)

Leading Data Mining Tools

- **C5.0/See5**, RuleQuest Research (Australia)
 - decision tree construction and rule induction
 - based on best-known data mining techniques
- **Intelligent Miner** family, IBM (USA)
 - customer relationship marketing; fraud and abuse detection
 - database support
- **Clementine**, Integral Solutions (UK)
 - neural networks and rule induction
 - data visualization in tables, histograms, plots and “webs”
- **Enterprise Miner**, SAS Institute (USA)
 - clustering, decision trees, & neural networks
 - GUI designed for business users

Leading Data Mining Tools (2)

- **DBMiner**, DBMiner Technology (Canada)
 - data warehousing support
 - a comprehensive set of mining facilities
- **Darwin**, Thinking Machines (US, distributors in Japan)
 - parallel, scalable, data mining architecture
 - NN, CART, and GAs
- **Pattern Recognition Workbench**, Unica (USA)
 - pattern classification and function estimation with an experiment manager
 - statistical regression, non-parametric, and NN algorithms

Leading Data Mining Tools (3)

■ Other Data Mining Tools:

- <http://www.datamininglab.com/toolcomp.html>
 - A Comparison of Leading Data Mining Tools
 - Data Mining Tools for Fraud Detection
 - Fourteen Desktop Data Mining Tools
- Weka 3: Machine Learning Software in Java (<http://www.cs.waikato.ac.nz/ml/weka/>)

Outline

- Knowledge Discovery in Databases: An Overview
- Data Mining at Large: Knowledge Engineering Support
 - Intelligent Learning Database Systems
- [Dealing with Large Databases: Existing Data Mining Techniques](#)
- Aggregating Rules from Different Data Sources
- Concluding Remarks

How Large Is a Large Database for Data Mining?

- According to Provost and Kolluri (1999),
 - Machine Learning: 100,000 instances with a couple dozen features, or more
 - Databases: 100 gigabytes, or larger (difficult to be processed simultaneously)
 - KDD (from an algorithmic perspective): one million examples (100 M byte – 1 G byte range)
- Dealing with large databases is a defining challenge in data mining research and development.

Dealing with Large Databases

- **Sampling:** random, stratified, and progressive strategies.
- Quinlan's **windowing** technique (1983) and **integrative windowing** (Furnkranz 1998)
- **Incremental learning:** training data items one by one
- **Multi-layer incremental induction** (Wu and Lo 1998): Subsets (or samples) of a database are processed one by one
 - **Data partitioning**
 - **Generalization:** A set of rules is learned.
 - **Reduction:** Behavioral examples from another set of examples are derived.
 - From behavioral examples, generalization can extract new rules.
 - Successive applications of the above generalization-reduction process allow more accurate and more complex (because of disjunctive) rules to be discovered, by sequentially handling the subsets of examples

Dealing with Large Databases (2)

- **Incremental batch learning** [Clearwater et al 1989]:
 - Rules are generated on each batch of examples, "almost-good-rules" are collected, an additional filter (such as Gen-Spec method and the n -Best) is applied, and then a RL technique (derived from the Meta-DENDRAL) is used to refine the remaining rules on the next batch of examples.
- **Meta-learning** for scalable inductive learning [Chan 1997]:
 - Partition a large database into subsets, run a learning algorithm on each of the subsets, and combine the results in some principled fashion. Since the learning processes are independent, they can be run in parallel for speed up over a sequential system.
 - Arbiters and combiners

Dealing With Large Databases (3)

- Learning "ensembles of classifiers" (such as **bagging** and **boosting**): Generates multiple versions of a predictor by running an existing learning algorithm many times on a set of re-sampled data. A data item in the original database can be used in many samples for generating different versions of the predictor.
- **Stacked generalization** [Wolpert 1992, Ting & Witten 1997] uses a high-level model to combine lower-level models to achieve greater predictive accuracy. These lower-level models are generated also by sampling the original database, and cross-validation is used to generate higher-level data for the high-level model.

Outline

- Knowledge Discovery in Databases: An Overview
- Data Mining at Large: Knowledge Engineering Support
 - Intelligent Learning Database Systems
- Dealing with Large Databases: Existing Data Mining Techniques
- [Aggregating Rules from Different Data Sources](#)
- Concluding Remarks

Aggregating Rules from Different Data Sources

- **Joint work with Shichao Zhang at Univ. of Technology, Sydney**
- Some databases are distributed, and/or are too large (e.g., with terabytes of data) to be processed at one time.
- Local rules at each data source may be useful for the local data in the first instance.
- Our approach: Collect rules from different data sources (or partitions), and synthesize these rules by weighting.

Association Analysis: An Example

$\text{supp}(A \rightarrow B)$: The fraction of transactions containing $A \cup B$;

$\text{conf}(A \rightarrow B)$: $\text{supp}(A \cup B) / \text{supp}(A)$

Suppose: $\text{minsupp} = 0.6$, and $\text{minconf} = 0.85$,
Valid rules: $a \rightarrow b$ and $d \rightarrow b$.

Transaction	Itemset
T1	{a,b,d}
T2	{a,b,d}
T3	{b,c,d}
T4	{b,c,d}
T5	{a,b}

Weight Allocation: An Example

(1) Rules from data source D1:

$A \cup B \rightarrow C$ with $\text{supp}= 0.4$, $\text{conf}=0.72$; (R1)

$A \rightarrow D$ with $\text{supp}= 0.3$, $\text{conf}=0.64$; (R2)

$B \rightarrow E$ with $\text{supp}= 0.34$, $\text{conf}=0.7$; (R3)

(2) Rules from data source D2:

$B \rightarrow C$ with $\text{supp}= 0.45$, $\text{conf}=0.87$; (R4)

$A \rightarrow D$ with $\text{supp}= 0.36$, $\text{conf}=0.7$; (R2)

$B \rightarrow E$ with $\text{supp}= 0.4$, $\text{conf}=0.6$; (R3)

(3) Rules from data source D3:

$A \cup B \rightarrow C$ with $\text{supp}= 0.5$, $\text{conf}=0.82$; (R1)

$A \rightarrow D$ with $\text{supp}= 0.25$, $\text{conf}=0.62$; (R2)

Summary: R1 supported by 2 data sources (D1 and D3);

R2 supported by 3 data sources (D1, D2 and D3);

R3 supported by D2 and D3; and R4 only by D2

Highest-belief rule: R2 (by its frequency).

Weight Allocation: Weights for Rules and Data Sources

Using the frequency of a rule as its weight with normalization,

$$W_{R_i} = \frac{\text{frequency}(R_i)}{\sum_j \text{frequency}(R_j)}$$

In the meanwhile, if a data source (D_i) supports a larger number of high-frequency rules, the the weight of D_i should also be higher.

$$W_{D_i} = \frac{\text{sum of } w_{R_k} * \text{frequency}(R_k) \text{ for all } R_k \text{ in } D_i}{\sum_j \text{sum } w_{R_j} * \text{frequency}(R_j) \text{ for all } R_j \text{ in } D_j}$$

Weight Aggregation

$$\text{supp}(X \rightarrow Y) = \sum_i w_i * \text{supp}_i(X \rightarrow Y)$$

$$\text{conf}(X \rightarrow Y) = \sum_i w_i * \text{conf}_i(X \rightarrow Y)$$

Aggregation of Association Rules from Different Data Sources

Input: D_i : a set of databases; $minsupp$, $minconf$: threshold values;

Output: S : a set of association rules;

1. **association analysis** with each database D_i ;
2. **assign** a weight to each rule (and possibly drop the low-belief rules with a pre-defined minimum frequency);
3. **assign** a weight to each D_i ;
4. **aggregate** all rules by weighting;
5. **rank** the rules;
6. **select** high rank rules to S , which have both support $\geq minsupp$ and confidence $\geq minconf$;
7. **output** S ;

Experiments

Data Sources: <http://www.kdnuggets.com/>

Database Name	Attributes	AvgAttr's/Row	Rows
T5I2.D100k	1000	5	100051
T10.I4.D100k	1000	10	98749
T20.I6.D100K	1000	20	99408

Results:

1. First 20 rules are **consistent** with Apriori when minsupp = 0.01, minconf = 0.65, and rules are ranked by their support.
2. **Time:** 1/10.

Outline

- Knowledge Discovery in Databases: An Overview
- Data Mining at Large: Knowledge Engineering Support
 - Intelligent Learning Database Systems
- Dealing with Large Databases: Existing Data Mining Techniques
- Aggregating Rules from Different Data Sources
- [Concluding Remarks](#)

Concluding Remarks

- The aggregation model provides a new, useful way to deal with large databases from different data sources.
- Classification of data sources before aggregation is sometimes necessary.
- More experiments (with real-world data) are required to test potential advantages of rule aggregation.