Relational Data Mining and GUHA

Tomáš Karban

DATESO 2005

April 14, 2005

Data Mining

- AKA knowledge discovery in databases
- Practice of automatic search for patterns in large data stores
  - implicit, previously unknown, interesting, potentially useful
- Techniques from statistics, machine learning, pattern recognition, propositional logic, ...

Taxonomy of Methods/Areas

- Classification/prediction
  - create a model from training data set and classify new examples (objects)
  - stress on accuracy
  - decision trees, decision rules, neural networks, Bayesian methods
- Descriptive methods
  - high level description, stress on simplicity
  - clustering methods
- Search for "nuggets"
  - interesting patterns, details, rules, exceptions, ...
  - mining for association rules

Single Table Limit

- Most methods use a single data table
  (data matrix, flat-file, attribute-value format)
  - rows = observations, objects, examples, items
  - columns = variables, properties, attributes, characteristics, features
- Real-world data usually stored in more data tables in relational database $\Rightarrow$ preprocessing to a single table
  - manual task, database joins, aggregations
  - more complex processing, e.g. time series analysis, linear regression, ...

Relational Data Mining

- Some methods or algorithms can be generalized to accept more data tables
  - relational classification rules, relational regression trees, relational association rules (WARMR)
- Methods of inductive logic programming (ILP) naturally use multiple data tables
- My doctoral thesis extends GUHA method for mining association rules from multiple data tables

Association Rules (1)

- Express relation between premise (antecedent) and consequence (succedent) $\varphi \approx \psi$
- φ and ψ are Boolean attributes derived as conjunctions from columns of studied data table
- $\approx$ stands for quantifier – truth condition based on contingency table of φ and ψ

- Example:

  Smoking(> 20cigs.) & PhysicalActivity(high) $\Rightarrow_{85\%}$ RespirationTroubles(yes)

Association Rules (2)

- Contingency table
- Founded implication $\Rightarrow_{p,Base}$

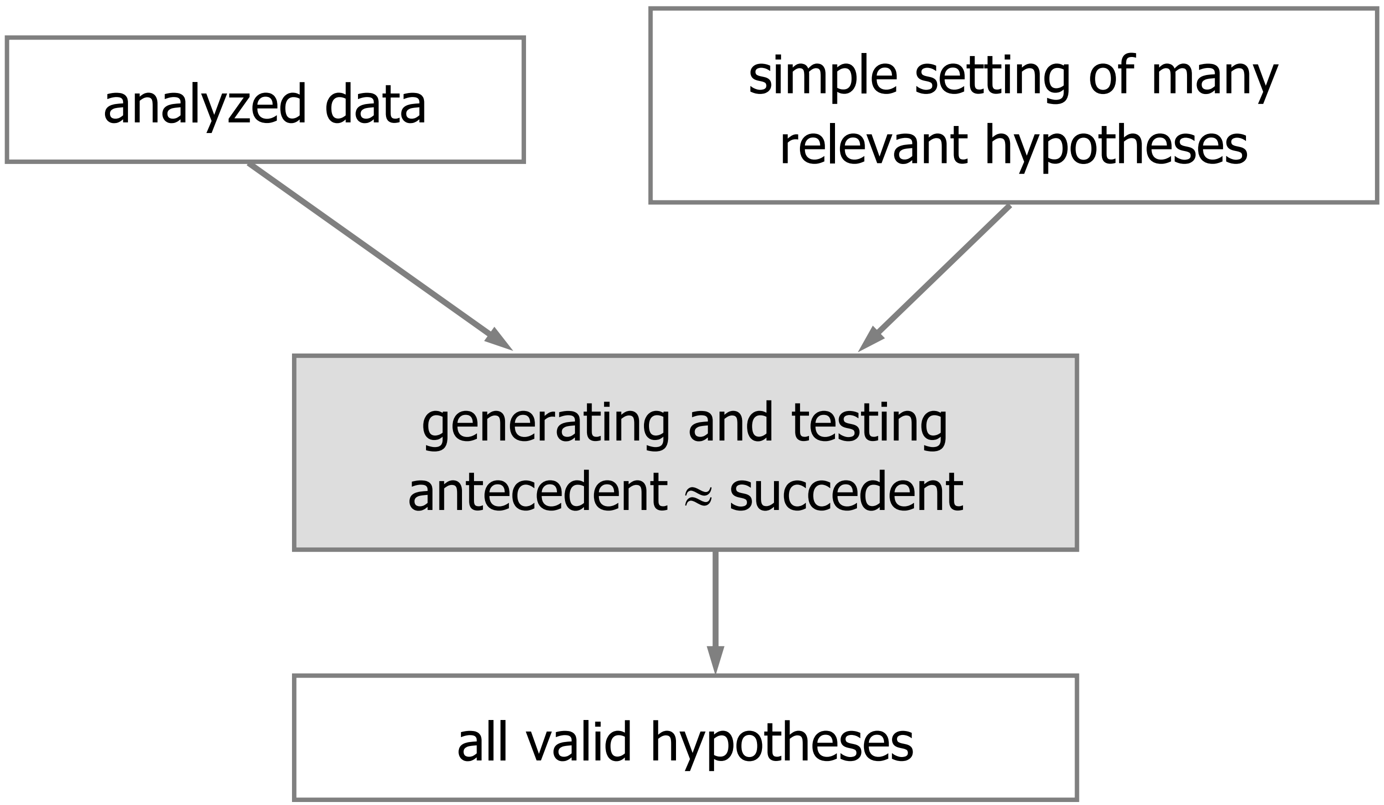
	ψ	$\neg\psi$
φ	a	b
$\neg\varphi$	c	d

$$\frac{a}{a+b} \geq p \quad \& \quad a \geq Base$$

- Various quantifiers available:
implications, double implications, equivalence, statistical hypotheses tests, above/outside average relations, etc.

GUHA Method

- Hájek, P. – Havránek, T.: *Mechanizing Hypothesis Formation – Mathematical Foundations for a General Theory*. Springer-Verlag, 1978



Effective Implementation

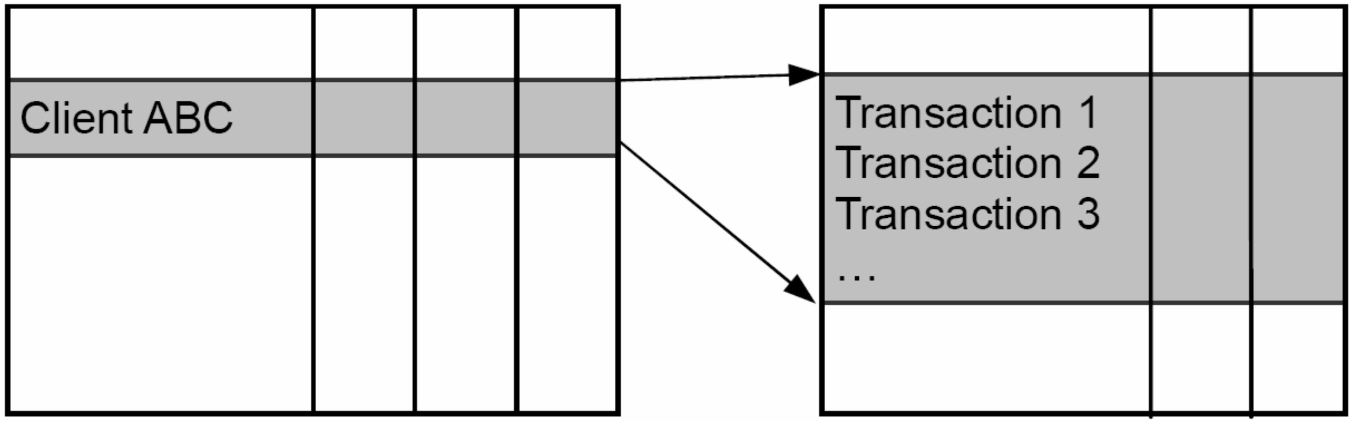
- Database is represented "vertically" in bit strings
  - bit string represents a single value of a single attribute
  - bit 1 denotes object has that value, bit 0 otherwise
- Antecedent, succedent are constructed as conjunction of literals (attributes or their negation)
  - using bitwise operations AND, NOT, OR
- Frequencies in contingency table are counts of 1 bits in bit strings $B\varphi \dot{\wedge} B\psi$, $B\varphi \dot{\wedge} B\neg\psi$, ...
- Careful handling of missing information (negation, three-valued logic)

An Alternative - APRIORI

- Aggraval, R. et al.: *Fast Discovery of Association Rules*. In Fayyad, U.M. et al.: Advances in Knowledge Discovery and Data Mining, pp. 307-328, AAAI Press / MIT Press, 1996
- Useful for market basket analysis (sparse data matrix)
- Transaction containing items A, B, C
  tend to contain item X as well (ABC $\rightarrow$ X)
  - measures: confidence, support
- Two phases
  - generating frequent itemsets
  - generating of association rules

Relational Association Rules

- We consider one data table as “the main”
- Additional tables are in 1:N relation
  - foreign key constraint, “master-detail”, star schema



- Clients: Birth, Gender, MaritalStatus, Children, LoanQuality
- Transactions: Date, TransactionAmount, SourceAccount, TargetAccount

Example

- MaritalStatus(divorced) & Children(3) & SingleIncome(yes) & AvgIncome(< 1500) $\Rightarrow_{76\%}$ LoanQuality(bad)

- SingleIncome derived as:

  TransactionAmount(> 500) $\Rightarrow_{93\%}$ SourceAccount(acc345) / Client(ABC)

  yes = strength of the hypothesis is greater than 90%

- AvgIncome derived as:

  AVG(SELECT SUM(TransactionAmount)
  WHERE (TransactionAmount > 0) GROUP BY YearMonth)

Adaptation to Relational DM

- Single table DM can be described by CRISP-DM methodology
  - ..., data preprocessing, modeling, ...
- Usually spiral development
  - after some success in modeling and evaluation, data are modified, prepared better, new run, ...
- Before-distinct steps now partially blend
  - some preprocessing is now given as a part of modeling setting and can be done semi-automatically (heuristics)

Virtual Attributes

- Basic notion is to bring data of some form from detail tables to main data table = create virtual attributes
- Three types:
  - aggregate attributes
  - existential attributes
  - association attributes (hypothesis attributes)
- In ILP world this is called "propositionalization"

WARMR

- Extension to APRIORI: Itemsets $\rightarrow$ Atomsets
  - existentially qualified conjunction (Prolog query)
  - frequent atomsets
  - + user-specified theory for pruning the search space

- Example:

  likes(K, dogs) & has(K, A) $\Rightarrow$ prefers(K, dogs, A)

  If child K likes dogs and already has an arbitrary animal A, he/she definitely prefers having dogs over A.

Comparison of GUHA and WARMR

- WARMR belongs to "selective methods" because of use of existentially qualified queries
  - suitable for structurally complex domains, e.g. molecular biology ("simple" data types, many tangled data tables)
  - association rules are structural patterns spanning many tables
- Rel-Miner belongs rather to "aggregating methods"
  - existential attributes are not so powerful,
they are limited to one detail table
  - suitable for non-determinate domains, usually in business (many-valued categories, real numbers, simple database schema)
  - association rules are focused on master table
which is enhanced by virtual attributes

Complexity of Relational Hypotheses

- Relational hypothesis space is enormous
  - it grows exponentially with the number of attributes (and their values)
  - number of virtual attributes is a sum of
    - meaningful aggregation attributes (low)
    - potentially useful association attributes
      - total number is exponential with the number of attributes in detail table, which is too much
      - potentially useful = hypothesis is true for some part of objects (say between 10% and 90%)
- Complex hypotheses are hard to interpret
  - they are not "interesting" in a sense...

Reordering the Verification

- We give up the idea that the whole hypothesis space can be crawled and verified
- Start with simplest hypotheses, go to more details
  - hypothesis complexity is vague
    - number of literals, user-defined importance of attributes
  - possible user interaction
    - interestingness of intermediate results, slight run-time modification of data mining task, user hints

Distributed Computing

- One database, one data preparation engine
- Many data mining processors
- Task can be split to disjoint fragments (jobs)
  - visual projection of hypothesis space = high-dimension cube
  - dimensions = attributes
  - fragments can be slices or mini-cubes
  - the whole task cube is “hollow” because of the limit on hypothesis length
- We can optimize task fragments to
  - take small amount of input (low number of bit strings)
  - be computed optimally (common sub-expressions in hypotheses)

Amount of Output

- Usual drawback of association rules = too many hypotheses as result
- User usually sorts them by some criteria that can be expressed as a real number
  - Adopting "TOP100" strategy, i.e. we can let the task to self-modify as we have some intermediate results
- Visualization - graph of hypotheses lattice
  - nodes = hypotheses, fuzzy edges = similarity of hypotheses

Conclusion

- New data mining tool Rel-Miner is being developed
- Builds on top of success of LISp-Miner
- It is different from ILP approach
  - aggregations
  - more expressive rules and quantifiers
  - slightly different target application domain
  - heuristics to deal with enormous hypothesis space

- Thank you!