



Database Mining: Bringing Algorithms to Data

Sharma Chakravarthy

Information Technology Laboratory
Computer Science and Engineering Department
The University of Texas at Arlington, Arlington, TX 76009
Email: sharma@cse.uta.edu
URL: <http://itlab.uta.edu/sharma>

Tutorial Outline

- Data Mining Overview
- Data Mining and DW
- Database Mining
 - Overview
 - Spectrum
- Database Mining Architectures
- Mining Using SQL
 - Association Rules
 - Graph Mining
 - Significant Interval Discovery
 - Event Pattern Discovery
- Summary and Challenges
- References



Tutorial Outline

- Data Mining Overview
- Database Mining
 - Need
 - Spectrum
- Database Mining Architectures
- Mining Using SQL
 - Association Rules
 - Graph Mining
 - Significant Interval Discovery
 - Event Pattern Discovery
- Summary and Challenges
- References



Motivation

Fraud division, some large telephone company:

"How do we find these guys? There are 10 billion records on 10 million customers in the main database. With all this information we have about our customers and all the calls they make, can't you just ask the database to figure out which lines have been set-up temporarily and exhibited similar calling patterns in the same time periods? The information is in there, I just know it ..."



Problem

- “Find-similar” problem just described is hard
 - e.g., “What products need to be improved?”
 - e.g., “Which books won’t be checked out and can be taken off the shelves?”

Why?

- Massive amounts of data
 - More and more online data stores (e.g., Web, click streams, corporate databases, etc.)
- No easy way to describe what to look for
- Traditional, interactive approaches fail
 - Size of data, different purposes



Another Example

- Marketing cellular phones
 - Churn is too high
 - Turnover after the initial contract is too high
 - What is a good strategy
 - Giving new phone to everyone is too expensive (and wasteful)
 - Bringing back customers after they leave is very difficult



What to do

- A few months before the contract expires, if one can predict which customers are likely to quit,
 - Give incentive to those who are likely to quit
 - Don't do anything for those who are NOT likely to quit
- How do I predict future behavior???
- Corporate Palm reading !
- Human intuition !!
- Data mining (DM) or knowledge discovery (KDD)



Data Mining

- *Data Mining* (DM) is part of the knowledge discovery process carried out to extract valid patterns and relationships in very large data sets
 - Usually don't know what to look for, like a “voyage into the unknown”
- Regarded as unsupervised learning from basic facts (axioms) and data
- Roots in AI and statistics
 - Uses techniques from machine learning, pattern recognition, statistics, database, visualization, etc.



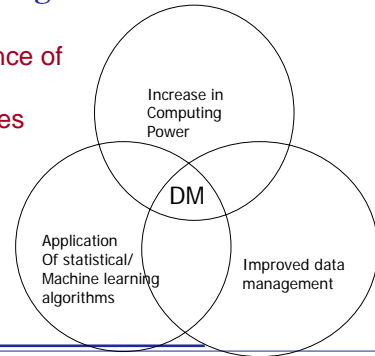
Enablers

- Reduced cost of storage
- Reduced cost of processing
- Ability to store, process, and manage large volumes of data
- New techniques such as association rules, sequence data processing, text mining
- However,
 - Scalability, visualization of results, filtering very large outputs are new issues!



Data Mining has come about due to

- Convergence of multiple technologies



Drivers

- Today, business information (or BI) systems are as important to corporations as transaction systems were earlier
- Mass personalization and better utilization of data
- Identify new and profitable markets, and channels to enter them
- Increase customer loyalty, profitability, life time value
- Decrease risk



AI and Statistics

- If DM is rooted in AI and statistics, what is new about it?
 - AI traditionally dealt with small samples
 - The emphasis was an learning, extrapolation, and generalization
 - The emphasis in DM is on processing the **actual data**, not just samples!
 - DM tries to leverage the data collected, accumulated and derive tangible rules/conclusions (generalization is also possible)



Machine Learning

- Observation
- Analysis
- Theory
- Prediction

Either the predictions are correct in which case the theory is corroborated, or the predictions are wrong.

New theory or exceptions!



DM Vs. Machine learning

- ML methods form the core of DM
- Amount of data makes a difference
 - accessing examples can be a problem
 - missing values and incomplete data
- DM has more modest goals: automating the tedious discovery tasks



DM Vs. Statistics

- Similar goals; additional methods
- Amount of data
- DM as a preliminary stage for statistical analysis
- Challenge to DM: better ties with statistics



Data Mining is NOT ...

- Data warehousing
- Ad hoc query/reporting
- Online Analytical Processing (OLAP)
- Data Visualization
- Agents/mediators,
- Pervasive computing, ...



What DM will NOT do !

- Substitute for human intuition and discovery
- I don't think a DM system will (ever?) discover $e = mc^2$
- I don't think DM will (ever?) discover $PV = RT$
- I don't think DM will (ever?) discover gravity, Newton's law's of motion
- On the other hand, It may discover new black holes !



PAKDD'06 Tutorial: SC
4/11/2006

Slide 17

Applications

- Customer profiling
 - Find new customers,
- *Market basket analysis*
 - Manage inventory, transportation, ...
- Risk analysis
 - Insurance, loan, stock, ...
- Text analysis
 - Library, Web, ...
- Fraud detection
 - Financial transactions, social networks
- CRM, Scientific discovery, forecasting, ...



PAKDD'06 Tutorial: SC
4/11/2006

Slide 18

DM Applications Vs. DM

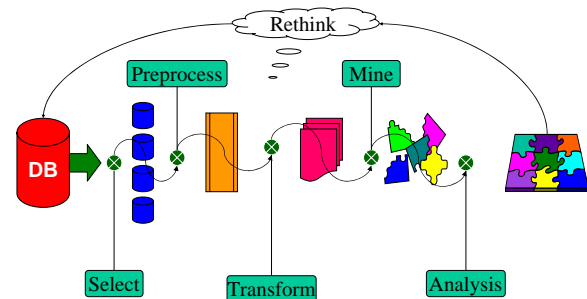
- Problem, goal and task definition (10%)
- Data Warehousing: data collection and organization (50%)
- Data Mining: data analysis and knowledge discovery (30%)
- Decision support / optimization: assess pros and cons, take actions (10%)



PAKDD'06 Tutorial: SC
4/11/2006

Slide 19

Data Mining Cycle



PAKDD'06 Tutorial: SC
4/11/2006

Slide 20

Common Pitfalls

- Misinterpretation of results
- Statistical significance
- Dirty data
- Too much information generated
- Legality
- Privacy/Ethics



Tutorial Outline

- Data Mining Overview
- **Data Mining and DW**
- Database Mining
 - Need
 - Spectrum
- Database Mining Architectures
- Mining Using SQL
 - **Association Rules**
 - **Graph Mining**
 - *Significant Interval Discovery*
 - *Event Pattern Discovery*
- Summary and Challenges
- References



Role of Data Warehouses

- DW makes DM a lot cheaper
- DM is one of the reasons for DW
- OLAP: verification-driven
 - sales in CA Vs. FL in Q1 of 2003
- DM: discovery-driven
 - What factors contribute to non-payment of loans ?
 - Will Microsoft come back?



OLAP Vs. Data Mining

- OLAP is *user driven*
 - Analyst generates hypothesis, uses OLAP to *verify*
 - e.g., "people with high debt are bad credit risks"
- Data mining tool *generates the hypothesis*
 - Tool performs exploration
 - e.g., find risk factors for granting credit
 - Discover new patterns that analysts didn't think of
 - e.g., debt-to-income ratio
- OLAP and DM complement each other



Tutorial Outline

ITL:AB
(U)
CSE:UDI

- Data Mining Overview
- Data Mining and DW
- Database Mining
 - Need
 - Spectrum
- Database Mining Architectures
- Mining Using SQL
 - Association Rules
 - Graph Mining
 - Significant Interval Discovery
 - Event Pattern Discovery
- Summary and Challenges
- References



PAKDD'06 Tutorial: SC
4/11/2006

Slide 25

Why Database Mining?

ITL:AB
(U)
CSE:UDI

- Proliferation of relational DW and the need to mine them **without siphoning** the data out
- Data mining must “co-exist” with OLAP and other decision-support applications
- DM need to be a sub-process in next generation Business Intelligence (BI) Systems
- Leverage the RDBMS technology for mining
- Provide an integrated decision-support environment for analysts



PAKDD'06 Tutorial: SC
4/11/2006

Slide 26

Data Mining Vs. Database Mining

ITL:AB
(U)
CSE:UDI

- Data mining refers to main memory algorithms for mining
 - + Can use arbitrary data structures
 - + Can optimize algorithms with proper representation (hash tree for example)
 - Limited memory, need for buffer management
 - Data has to be siphoned out of its location (mostly a DBMS or a Data Warehouse)
 - Works well only for small data sizes (no scalability)
 - Every time data is added to the DB, the process has to be repeated

Solution? Database Mining – Bringing algorithms to data instead of taking data to algorithms



PAKDD'06 Tutorial: SC
4/11/2006

Slide 27

SQL-based Mining: Implications

ITL:AB
(U)
CSE:UDI

- Leverage 2+ decades of DBMS R&D
- Portability due to standardization
- Fast development of mining algorithms
- SMP parallelism for free for parallel database engines
- Data is not replicated outside of DBMS
- SQL may be extended to include *ad hoc* mining queries
- However, No specialized data structures and memory management



PAKDD'06 Tutorial: SC
4/11/2006

Slide 28

Data Mining Evolution

ITL:AB
(U)
CSE:U:DI

- **File-Based** or Main Memory mining algorithms
 - Data mining
- **SQL-Based** mining algorithms
 - Database mining
- **Parallel mining Algorithms**
 - Both main memory and SQL-based



PAKDD'06 Tutorial: SC
4/11/2006

Slide 29

Tutorial Outline

ITL:AB
(U)
CSE:U:DI

- Data Mining Overview
- **Database Mining**
 - Need
 - **Spectrum**
- Database Mining Architectures
- Mining Using SQL
 - **Association Rules**
 - **Graph Mining**
 - *Significant Interval Discovery*
 - *Event Pattern Discovery*
- Conclusions
- Summary and Challenges
- References



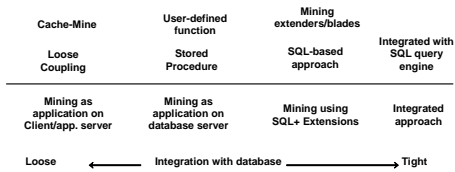
PAKDD'06 Tutorial: SC
4/11/2006

Slide 30

Mining Spectrum

ITL:AB
(U)
CSE:U:DI

- Study architectural alternatives
- Performance evaluation
- Extend the capability of current query processors



PAKDD'06 Tutorial: SC
4/11/2006

Slide 31

Database Mining Spectrum

ITL:AB
(U)
CSE:U:DI

- **Database Mining**
 - Single database (Directly) e.g. Intelligent Miner
 - Single relation (using JDBC)
 - Layered (multiple relations, using JDBC)
 - Layered (Across databases, using JDBC)
 - Integrated Database mining



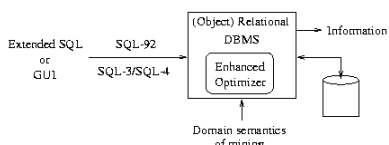
PAKDD'06 Tutorial: SC
4/11/2006

Slide 32

Long-Term Vision

FLAB
CSE/UDI

- Unbundle bulky mining operations
- Identification of Common operators
- Integration of the above into the Query Optimizer
- No distinction between OLAP and mining



PAKDD'06 Tutorial: SC
4/11/2006

Slide 33

Tutorial Outline

FLAB
CSE/UDI

- Data Mining Overview
- Database Mining
 - Need
 - Spectrum
- Database Mining Architectures
- Mining Using SQL
 - Association Rules
 - Graph Mining
 - Significant Interval Discovery
 - Event Pattern Discovery
- Conclusions
- Summary and Challenges
- References



PAKDD'06 Tutorial: SC
4/11/2006

Slide 34

Alternatives Architectures

FLAB
CSE/UDI

- Loose-coupling: data read through a cursor
- Stored-procedure: mining algorithm encapsulated as a stored procedure (SP)
- Cache-mine-store: data cached in files outside DB in binary form
- UDF: "heavy-weight" UDFs placed in SQL queries
- SQL: mining algorithm formulated as SQL-92/SQL-OR queries



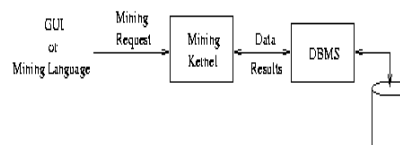
PAKDD'06 Tutorial: SC
4/11/2006

Slide 35

Loosely Coupled

FLAB
CSE/UDI

- Loose-coupling: data read through a cursor
- Intermediate results are stored in the database
- DBMS used as file system
- High context switch between address spaces
- Even with block reads, performance is poor



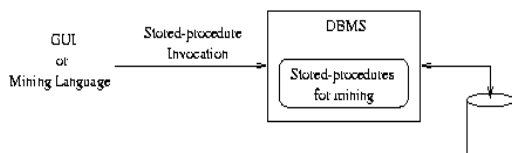
PAKDD'06 Tutorial: SC
4/11/2006

Slide 36

Stored procedures

FLAB
CSE-UDI

- Mining algorithms executed as Stored-Procedures on the server
- Programming flexibility
- Existing file code can be reused



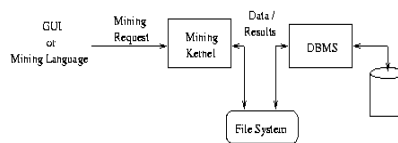
PAKDD'06 Tutorial: SC
4/11/2006

Slide 37

Cache-Mine-Store

FLAB
CSE-UDI

- Data is read once and cached in files outside DB (in binary form)
- Advantages of SP + better performance
- Additional disk space for caching



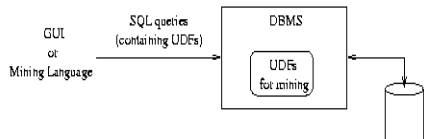
PAKDD'06 Tutorial: SC
4/11/2006

Slide 38

UDFs

FLAB
CSE-UDI

- UDF: "heavy-weight" UDFs placed in SQL queries
- Little use of DB query processing
- Fenced or unfenced mode
- Performance is good
- Portability is poor



PAKDD'06 Tutorial: SC
4/11/2006

Slide 39

Stored Procedures and UDFs

FLAB
CSE-UDI

- Stored procedures and UDFs on the server side
- Advantages
 - Less traffic congestion
 - Better development
 - Modularization and Integration
 - Can return from basic data types like integers to complex structures like tables
- Client side - coding of these functions takes time and effort



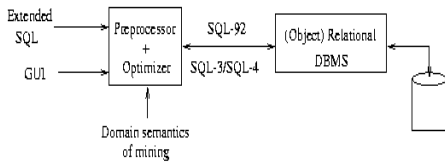
PAKDD'06 Tutorial: SC
4/11/2006

Slide 40

SQL-Based

FLAB
CSE@UDI

- SQL: mining algorithm formulated as SQL-92/SQL-OR queries
- Several alternatives (query/subquery, Kway join)
- Exploit SQL parallelism



PAKDD'06 Tutorial: SC
4/11/2006

Slide 41

Tutorial Outline

FLAB
CSE@UDI

- Data Mining Overview
- Database Mining
 - Need
 - Spectrum
- Database Mining Architectures
- Mining Using SQL
 - Association Rules
 - Graph Mining
 - Significant Interval Discovery
 - Event Pattern Discovery
- Summary and Challenges
- References



PAKDD'06 Tutorial: SC
4/11/2006

Slide 42

Association Rules

FLAB
CSE@UDI

- To discover associations, we assume that we have a set of transactions, each transaction being a list of items (e.g., list of books, items bought from a store)
- Suppose A and B appear together in only 1% of the transactions but whenever A appears there is 80% chance that B also appears
- The 1% presence of A and B together is called the support of the rule and 80% is called the confidence of the rule ($A \rightarrow B$)



PAKDD'06 Tutorial: SC
4/11/2006

Slide 43

Association Rule Mining

FLAB
CSE@UDI

Customer transactions

Tid Item	Bread	Sugar	Eggs	Milk	Cerea
1	X		X	X	
2		X	X		X
3		X	X	X	X
4		X			X

Rules

Head	Symbol	Body	Support	Confidence
Sugar	\Rightarrow	Cereal	75.00	100.00
Cereal	\Rightarrow	Sugar	75.00	100.00
Eggs	\Rightarrow	Milk	50.00	66.67



PAKDD'06 Tutorial: SC
4/11/2006

Slide 44

Association Rules: Details

ITL:AB
(U)
CSE:U:DI

- Consider the rule $A \rightarrow B$.
- Suppose A and B together appear in 10 out of 100 transactions.
 - Then support is 10/100 or 10%
- Suppose A by itself appears in 20 out of 100 transactions.
 - That means that A appears in 20 transactions and (A, B) appear in 10 transactions.
 - Confidence is 50% or 10/20



PAKDD'06 Tutorial: SC
4/11/2006

Slide 45

Association Rule Mining

ITL:AB
(U)
CSE:U:DI

- Consider a set of transactions, where in a transaction, a customer purchases a number of items.
 - Association rules are of the form $X \Rightarrow Y$.
 - X and Y are set of items bought in the same transaction, such that $X \cap Y = \Phi$
 - Support and Confidence:
 - **Support** $\{ X \Rightarrow Y \} =$
$$\frac{\text{Number of Transactions containing itemset } X \cup Y}{\text{Total Number of Transaction}}$$
 - **Confidence of the rule** $\{ X \Rightarrow Y \} =$
$$\frac{\text{Support of } \{ X \cup Y \}}{\text{Support } \{ X \}}$$



PAKDD'06 Tutorial: SC
4/11/2006

Slide 46

Association Rules

ITL:AB
(U)
CSE:U:DI

- Capture co-occurrence of items/events
- Discover all rules with minimum support and conf.

Approach:

- Find all the frequent itemsets
- Generate rules from the frequent itemsets

Apriori Algorithm:

- Level-wise approach involving multiple data passes
- Candidate generation, pruning,
- Support counting



PAKDD'06 Tutorial: SC
4/11/2006

Slide 47

Association Rule Algorithm

ITL:AB
(U)
CSE:U:DI

- To find such associations, a simple two step approach may be used:
- Step 1 - discover all **frequent items** that have support above the minimum support required
- Step 2 - Use the set of frequent items to generate the association rules that have high enough confidence



PAKDD'06 Tutorial: SC
4/11/2006

Slide 48

The Apriori Algorithm

- The algorithm works as follows:
- Scan all transactions and find all items that have transaction support above $x\%$. Let these be F_1 .
- Build item pairs from F_1 . This is the candidate set C_2 . Scan all transactions and find all frequent pairs in C_2 . Let this be F_2 .
- General rule - build sets of k items from F_{k-1} . This is set C_k . Scan all transactions and find all frequent sets in C_k . Let this be F_k .



The Apriori Algorithm

```

F(1) = {Frequent single itemsets};
for (k=2 ; F(k-1)≠∅ ; k++) do begin
  C(k) = Apriori-generation(F(k-1)); // new candidates
  for each transaction t ∈ D begin
    C(t) = subset(C(k),t); // Candidates contained in t
    for each candidate c ∈ C(t) do
      c-count ++;
  end;
  F(k) = {c ∈ C(k) | c-count ≥ minimum support}
end
return  $\cup_k F(k)$  ;

```



Apriori candidate generation (join step)

- The *Apriori-generation* function takes as argument $F(k-1)$, the set of all frequent $(k-1)$ -item sets. it returns a superset of the set of all frequent k -item sets. The function works as follows: First, in the join step, we join $F(k-1)$ with $F(k-1)$:

```

insert into C(k)
select p.item(1), p.item(2),... p.item(k-1), q.item(k-1)
from F(k-1) as p, F(k-1) as q
where p.item(1) = q.item(1),...,p.item(k-2) = q.item(k-2),
p.item(k-1) < q.item(k-1)

```



The prune step

- we delete all the item sets c in $C(k)$ such that some $(k-1)$ -subset of c is not in $F(k-1)$:
for each item sets c in C(k) do
for each (k-1)-subsets s of c do
if (s not in F(k-1)) then
delete c from C(k);
- Any subset of a frequent item set must be frequent
- Lexicographic order of items is assumed!
- Apriori is using the *monotonicity property*
 "all subsets of a frequent itemset are also frequent" for pruning.
 However, NOT all supersets of a frequent itemset are frequent.





SQL-92 based approaches to Association Rules

Implementation on Oracle and
IBM DB2/UDB

Input to Mining

TID	Item1	Item2	Item3	Item4	Item5
100	1		1	1	
200		1	1		1
300	1	1	1		1
400		1		1	



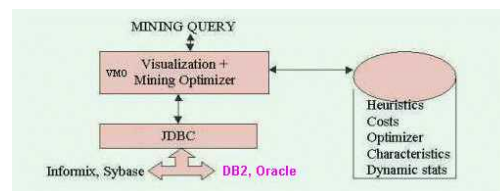
Input Table for Database Mining

TID	ITEM
100	1
100	3
100	4
200	2
200	3
200	5
300	1
300	2
300	3
300	5
400	2
400	5

Why is this representation
Used? And not the previous one?

Short-term Goal

- Layered architecture



- JDBC provides the database connection and SQL interface
- VMO generates and visualizes the association rules



SQL-based Support counting

FTL/AB
(U)
CSE/UDI

- SQL-92
 - K-way joins
 - Subquery
 - 3-way joins
 - 2- group by
- SQL-OR
 - GatherJoin
 - GatherPrune
 - GatherCount
 - SQL-bodied functions



PAKDD'06 Tutorial: SC
4/11/2006

Slide 57

Characteristics

FTL/AB
(U)
CSE/UDI

- Number of items : in thousands
- Number of Transactions: in Millions
- Data set sizes: High Gigabytes
 - Discovering all rules rather than *verifying* if a rule holds
 - Completeness
 - Performance
 - Scalability



PAKDD'06 Tutorial: SC
4/11/2006

Slide 58

Input/Output Formats

FTL/AB
(U)
CSE/UDI

- Input transaction table in the normal form
- Two attributes (tid, item)
- Example: 1: A, B, C
- Output is a collection of rules
- Rule table schema:
(item₁, ..., item_k, len, rulem, confidence, support)
- Rule AB -> CD, conf. 90%, support 5%
(A, B, C, D, NULL, 4, 2, 0.9, 0.05)

Tid	Item
1	A
1	B
1	C



PAKDD'06 Tutorial: SC
4/11/2006

Slide 59

K-way Join

FTL/AB
(U)
CSE/UDI

- The process of support counting in K_{wj} is as follows:
In any pass k:
 - Frequent itemsets of length k-1 are used to generate candidate itemsets of length k (C_k).
 - For support counting of these candidate itemsets, k copies of input relation is joined with the C_k.



PAKDD'06 Tutorial: SC
4/11/2006

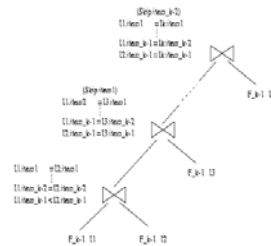
Slide 60

Candidate Generation in SQL

- Join step: join 2 copies of F_{k-1}

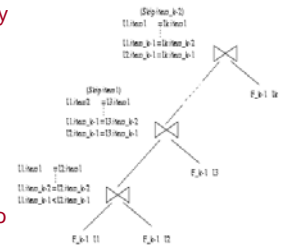
```

insert into Ck
select I1.item1, ..., I1.itemk-1, I2.itemk-1
from Fk-1 I1, Fk-1 I2
where I1.item1 = I2.item1 and .... and
I1.itemk-2 = I2.itemk-2 and
I1.itemk-1 < I2.itemk-1
    
```



Candidate Generation and Pruning

- Prune step: additional joins with $(k-2)$ more copies of F_{k-1}
- Join predicates enumerated by skipping an item at a time
- K items have $k(k-1)$ subsets; Out of that 2 have been used for generating the K item. No need to check them. Hence, the other $(k-2)$ subsets need to be checked by taking $(k-2)$ joins



Candidate Set C_k Generation

```

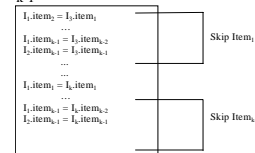
Insert into Ck
Select I1.item1, I1.item2, ..., I2.itemk-1, count(*)
From Fk-1 I1, Fk-1 I2
Where I1.item1 = I2.item1 AND
      I1.item2 = I2.item2 AND
      ...
      I1.itemk-2 = I2.itemk-2 AND
      I1.itemk-1 < I2.itemk-1
    
```

Example: $F_3: \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 3, 4\}\}$
 $\Rightarrow C_4: \{1, 2, 3, 4\}$, and $\{1, 3, 4, 5\}$.



Candidate Set C_k Generation

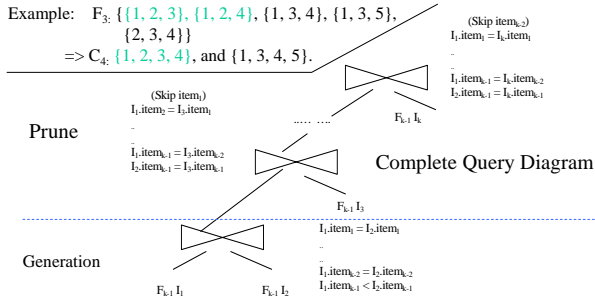
Prune step: in the k -itemset of C_k , if there is any $(k-1)$ -subset of C_k that is not in F_{k-1} , we need to delete that k -itemset from C_k .



In the above example, one of the 4-itemset in C_4 is $\{1, 3, 4, 5\}$. This 4-itemset needs to be deleted because one of the 3-item subsets $\{3, 4, 5\}$ is not in F_3 .



Candidate Set Ck Generation



Pruning explanation

- Consider $C_k \{1\ 3\ 4\ 5\}$
- The subsets are
 - $\{1\ 3\ 4\}$ generated by skipping item 4
 - $\{1\ 3\ 5\}$ generated by skipping item 3
 - $\{1\ 4\ 5\}$ generated by skipping item 2
 - $\{3\ 4\ 5\}$ generated by skipping item 1
- First 2 have been used in the generation of $\{1\ 3\ 4\ 5\}$
- Hence, skip 1 and 2 or 1 thru k-2 (here k is 4)

SQL-92 support counting - Kway

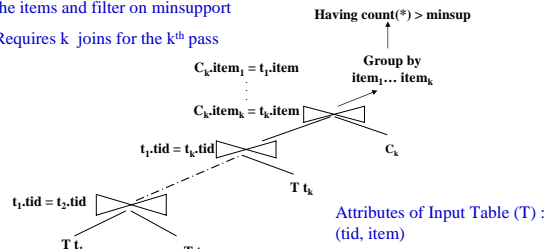
- Join k copies of input table with C_k and do a group by on the itemsets
- insert into F_k

```

select item1, ..., itemk, count(*)
from Ck, T1, ..., Tk
where t1.item = Ck.item1 and
...
tk.item = Ck.itemk and
t1.tid = t2.tid and
...
tk-1.tid = tk.tid
group by item1, item2, ..., itemk
having count(*) ≥ minsup
        
```

Support Counting for Kwj in any pass k

Join C_k with k copies of T
 Follow up the join with a group by on the items and filter on minsupport
 Requires k joins for the k^{th} pass



Example: Frequent itemsets generation using Kwj

FLAB
CSE-UDI

Tid	lid
1	1
1	3
1	4
2	2
2	3
2	5
3	2
3	3
3	4
3	5
4	2
4	5

Item	Count
2	3
3	3
4	2
5	3

Item	Count
2	3
3	3
4	2
5	3

Item ₁	Item ₂
2	3
2	4
2	5
3	4
3	5
4	5

Tid	lid
1	1
1	3
1	4
2	2
2	3
2	5
3	2
3	3
3	4
3	5
4	2
4	5

Tid	lid
1	1
1	3
1	4
2	2
2	3
2	5
3	2
3	3
3	4
3	5
4	2
4	5

Item1	Item2	Count
2	3	2
2	5	3
3	4	2
3	5	2

Item1	Item2	Item3
2	3	5
3	4	5

PAKDD'06 Tutorial: SC
4/11/2006

Slide 69

FLAB
CSE-UDI

SQL-92 support counting - Kway (Contd.)

- Simple and by far the best approach
- Fast on both the experimental databases
- Only down side is the use multiple binary joins

PAKDD'06 Tutorial: SC
4/11/2006

Slide 70

FLAB
CSE-UDI

Rule Visualization

Rule Table with *Filter* capability

PAKDD'06 Tutorial: SC
4/11/2006

Slide 71

The key point is to construct a *where* clause using the standard SQL operators, such as 'LIKE', 'NOT', 'IN', 'AND', etc

FLAB
CSE-UDI

Rule Visualization

Rule Table with *Sort* capability

PAKDD'06 Tutorial: SC
4/11/2006

Slide 72

Methodology for experiments

FLAB
@
CSE@UOI

- Synthetic data sets, generated by using IBM's data-generator.
- Datasets are named as TxxlyDzzzK.
 - xx denotes the average number of items present per transaction.
 - yy denotes the average support of each item in the dataset.
 - zzzK denotes the total number of transactions in K (1000's).
 - Example: T5I2D1000K.
- Tested on Oracle 8i and IBM DB2/UDB V6.1
- Each experiment has been performed 4 times in a row.
- Most results are shown for three datasets – T5I2D500K, T5I2D1000K and T10I4D100K.

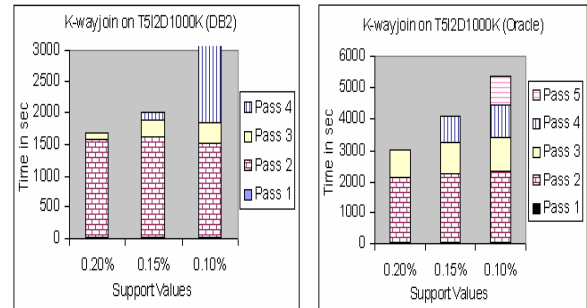


PAKDD'06 Tutorial: SC
4/11/2006

Slide 73

Experiments

FLAB
@
CSE@UOI



PAKDD'06 Tutorial: SC
4/11/2006

Slide 74

Observations

FLAB
@
CSE@UOI

- C₁ (typically input relation) can be substituted by F₁ for subsequent passes
- Second pass is the most expensive one; no pruning at all; it is a Cartesian product
- As the number of passes increases (i.e., longer frequent itemsets), the number of joins increases; hence materialization may be effective



PAKDD'06 Tutorial: SC
4/11/2006

Slide 75

Optimizations

FLAB
@
CSE@UOI

- Reduce the size of input dataset
 - Non-frequent 1-itemsets are pruned out from the input table and this pruned input table is used instead in further passes.
 - Effective for higher supports
- Optimize the second pass.
 - Skip generation of F₁ and C₂ and directly generate F₂ by joining 2 copies of input dataset.
 - Effective for all large data sets
- Reduce the number of joins done in any pass
 - Materialize all the frequent itemsets contained in any transaction at the end of the pass k and use them for support counting in pass k+1
 - Effective for higher iterations

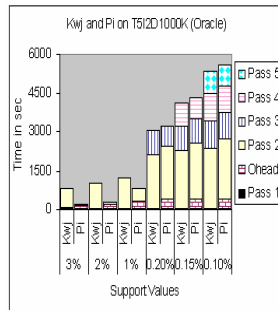
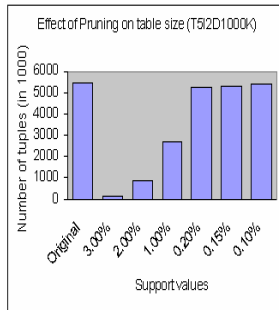


PAKDD'06 Tutorial: SC
4/11/2006

Slide 76

Experiments

FLAB
CSE-UDI



Effect of Pruning

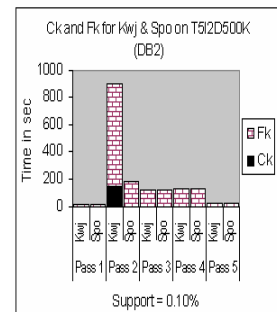
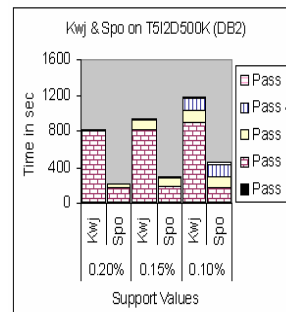


PAKDD'06 Tutorial: SC
4/11/2006

Slide 77

Experiments

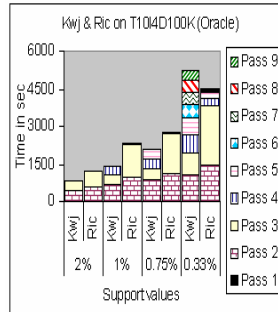
FLAB
CSE-UDI



PAKDD'06 Tutorial: SC
4/11/2006

Slide 78

- In k^{th} pass create a relation $\text{Comb}_k(\text{tid}, \text{item}_1, \text{item}_2, \dots, \text{item}_k)$.
- Join Comb_{k-1} with T and C_k and insert into Comb_k only those transactions from T that have candidate itemsets which are one extension to the candidate itemsets in Comb_{k-1} .
- Do a group by on Comb_k to generate F_k .
- Thus in any pass k , we have only 3 joins, instead of $k+1$ joins.
- Comb_k and comb_k are quite large; hence takes more time



PAKDD'06 Tutorial: SC
4/11/2006

Slide 79

Effect of these optimization

FLAB
CSE-UDI

- Pruned input**
 - Effective only when support value is high, otherwise at low support values the cost of pruning is more.
- Second Pass Optimization**
 - Effective in all cases
- Reuse of Item Combinations**
 - Effective when the number of passes is high otherwise the cost of materialization is more



PAKDD'06 Tutorial: SC
4/11/2006

Slide 80

Combinations of Optimizations

FILAB
CSE-UDA

- Second pass optimization on pruned input (SpoPi)
 - Good for higher values of support
- 1. Reuse of item combination on pruned input (RicPi)
 - Good for higher passes and large support; but larger support reduces the # of passes
- 2. Reuse of item combination with second pass optimization (RicSpo)
 - Good certainly for higher passes; good for any support
- 3. Combination of all the optimization (All)
 - Observed some differences between Oracle and DB2

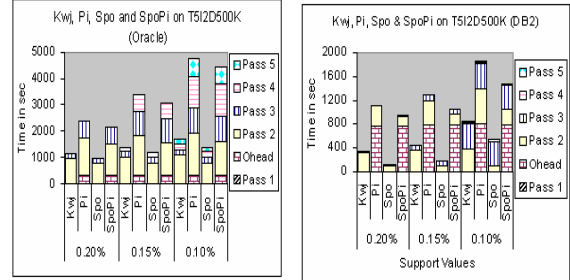


PAKDD'06 Tutorial: SC
4/11/2006

Slide 81

Second Pass Optimization on Pruned Input (SpoPi)

FILAB
CSE-UDA



Pi optimization for low values of support does not help as the cost of pruning needs to be included.

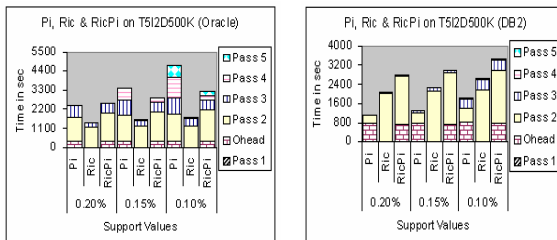


PAKDD'06 Tutorial: SC
4/11/2006

Slide 82

Reuse of Item Combinations on Pruned Input (RicPi)

FILAB
CSE-UDA



Reuse is good for higher passes and pruning is good for higher support
But higher support typically reduces the number of passes

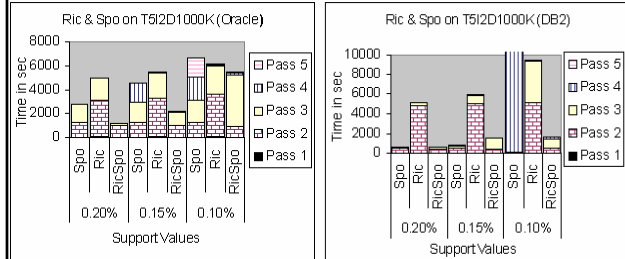


PAKDD'06 Tutorial: SC
4/11/2006

Slide 83

Reuse of Item Combinations and Second Pass Optimization (RicSpo)

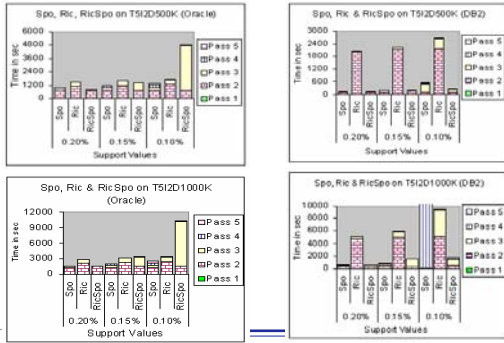
FILAB
CSE-UDA



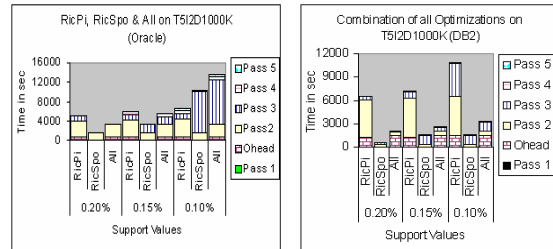
PAKDD'06 Tutorial: SC
4/11/2006

Slide 84

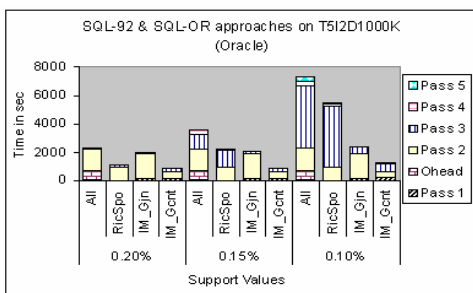
Reuse of Item Combinations and Second Pass Optimization (RicSpo)



Combination of all Optimizations



Comparison of SQL based approaches



Results

Table Name	Ranking	Supp = 0.2%	Supp = 0.15%	Supp = 0.1%	
T512D100K	First	RicSpo	RicSpo	Kwj	
	Second	All	All	RicSpo	
	Last	RicPi	RicPi	RicPi	
T512D500K	First	RicSpo	RicSpo	Spo	
	Second	Spo	Spo	RicSpo	
	Last	RicPi	RicPi	RicPi	
	Ranking	Supp = 2.0%	Supp = 1.0%	Supp = 0.75%	Supp = 0.33%
T1014D100K	First	All	RicSpo	RicSpo	
	Second	Pi	All	All	Spo
	Last	Ric	RicPi	RicPi	RicSpo

Trends in Oracle for SQL-92 based approaches



Results

FTL:AB
(U)
CSE:UDI

Table Name	Ranking	Supp = 0.2%	Supp = 0.15%	Supp = 0.1%
T512D100K	First	RicSpo	Spo	RicSpo
	Second	Spo	RicSpo	All
	Last	RicPi	SpoPi	SpoPi
T512D500K	First	Spo	Spo	Spo
	Second	RicSpo	RicSpo	RicSpo
	Last	SpoPi	SpoPo	SpoPi
	Ranking	Supp = 2.0%	Supp = 1.0%	Supp = 0.75%
T1014D100K	First	Spo	RicSpo	RicSpo
	Second	RicSpo	All	All
	Last	Ric	Kwj	Kwj

Trends in IBM DB2/UDB for SQL-92 based approaches



PAKDD'06 Tutorial: SC
4/11/2006

Slide 89

Metadata Table

FTL:AB
(U)
CSE:UDI

- Based on the cardinality.
- Underlying RDBMS.
- Whether we can use any extra space.



PAKDD'06 Tutorial: SC
4/11/2006

Slide 90

Summary Table for SQL-92 based Approaches

FTL:AB
(U)
CSE:UDI

T512DzzzK	IBM DB2/UDB		Oracle		Support Value
	Extra Space	No Extra Space	Extra Space	No Extra Space	
10K	RicSpo	Spo	RicSpo	Spo	S = 0.20 %
	RicSpo	Spo	RicSpo	Spo	S = 0.15 %
	RicSpo	Spo	Spo	Spo	S = 0.10 %
50K	RicSpo	Spo	RicSpo	Spo	S = 0.20 %
	Spo	Spo	RicSpo	Spo	S = 0.15 %
	Spo	Spo	Spo	Spo	S = 0.10 %
100K	RicSpo	Spo	RicSpo	Spo	S = 0.20 %
	Spo	Spo	RicSpo	Spo	S = 0.15 %
	Spo	Spo	Spo	Spo	S = 0.10 %
500K	Spo	Spo	RicSpo	Spo	S = 0.20 %
	Spo	Spo	RicSpo	Spo	S = 0.15 %
	Spo	Spo	Spo	Spo	S = 0.10 %
1000K	RicSpo	Spo	RicSpo	Spo	S = 0.20 %
	Spo	Spo	RicSpo	Spo	S = 0.15 %
	Spo	Spo	Kwj	Spo	S = 0.10 %



PAKDD'06 Tutorial: SC
4/11/2006

Slide 91

Summary of SQL92 Approaches for Association Rule Mining

FTL:AB
(U)
CSE:UDI

- Data Mining is a tool box consisting of different approaches useful for different data sets
- Choosing the appropriate approach is one of the important aspects of data mining
- Understanding the domain and matching the DM techniques for what one wants to do is another challenge
- Interpreting the results of the mining output is the third challenge



PAKDD'06 Tutorial: SC
4/11/2006

Slide 92

Tutorial Outline

ITL:AB
(U)
CSE:U:DI

- Data Mining Overview
- Database Mining
 - Need
 - Spectrum
- Database Mining Architectures
- Mining Using SQL
 - Association Rules
 - **Graph Mining**
 - *Significant Interval Discovery*
 - *Event Pattern Discovery*
- Summary and Challenges
- References



PAKDD'06 Tutorial: SC
4/11/2006

Slide 93

Graph Data Mining: Overview

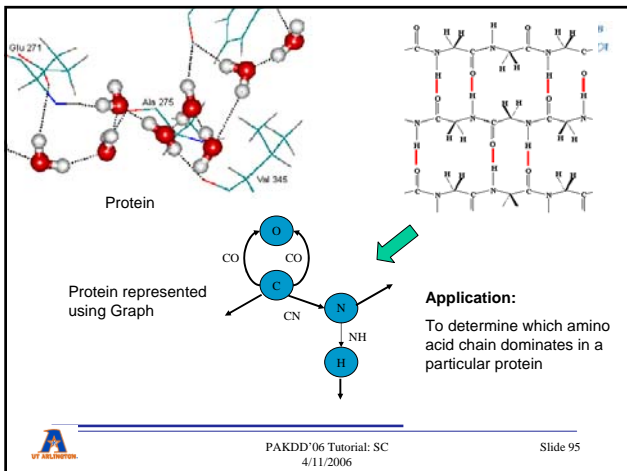
ITL:AB
(U)
CSE:U:DI

- Association Rule Mining, decision trees ... mine transactional data.
- Graph based mining techniques are used for mining data that are structural in nature (chemical compounds, complex proteins, VLSI circuits, ...), as mapping them to other representations will lead to the loss of structural information
- Significant work in the area includes the Subdue substructure discovery algorithm (Cook & Holder), HDBSubdue (Chakravarthy, Beer, Padmanabhan), the frequent subgraph (FSG) technique (Karypis & Kuramochi), and the gSpan approach (J. Han)



PAKDD'06 Tutorial: SC
4/11/2006

Slide 94



PAKDD'06 Tutorial: SC
4/11/2006

Slide 95

Application Domains

ITL:AB
(U)
CSE:U:DI

- Chemical Reaction chains
- CAD Circuit Analysis
- Social Networks
- Credit Domains
- Web analysis
- Games (Chess, Tic Tac toe)
- Program Source Code analysis
- Chinese Character data bases
- Geology
- Aviation Data Bases



PAKDD'06 Tutorial: SC
4/11/2006

Slide 96

Graph Based Data Mining

FTL/AB
(U)
CSE/UDI

- A Graph representation of the database is intuitive and an obvious choice.
- Graphs can be used to accurately model and represent scientific data sets. Graphs are suitable for capturing arbitrary relations between the various objects.

Data Instance	Graph Instance
Object	Vertex
Object's Attributes	Vertex Label
Relation Between Two Objects	Edge
Type Of Relation	Edge Label

- Graph based data mining aims at discovering interesting and repetitive patterns within these structural representations of data.



PAKDD'06 Tutorial: SC
4/11/2006

Slide 97

Graph Mining Overview

FTL/AB
(U)
CSE/UDI

- A substructure is a connected subgraph; need to differentiate between substructures and substructure instances
- A **connected subgraph** is a subgraph of the original graph where there is a path between any two vertices
- A subgraph $G_s = (V_s, E_s)$ of $G = (V, E)$ is **induced** if E_s contains all the edges of E that connect vertices in V_s
- **Directed** and **undirected** edges are needed; **multiple edges** between two nodes need to be accommodated; **cycles** need to be handled



PAKDD'06 Tutorial: SC
4/11/2006

Slide 98

Graph Mining: Complexity

FTL/AB
(U)
CSE/UDI

- Enumerating all the substructures of a graph has exponential complexity
- Subgraph isomorphism is NP complete
- Generating canonical labels is $O(|V|!)$, where V is the number of vertices
- All approaches have to deal with the above in order to be able to work on large data sets
- Different approaches do it differently; scalability depends on its and the use of buffers



PAKDD'06 Tutorial: SC
4/11/2006

Slide 99

Subdue

FTL/AB
(U)
CSE/UDI

- One of the **earliest work** in Graph based data mining
 - Uses **sparse adjacency matrix** for graph representation
- Substructures are evaluated using a metric called **Minimum Description Length** principle based on adjacency matrices
- Capable of matching two graphs, differing by the number of vertices specified by the threshold parameter, **inexactly**
- Performs **hierarchical clustering** by compressing the input graph with best substructure in each iteration



PAKDD'06 Tutorial: SC
4/11/2006

Slide 100

Motivation

ITLAB
CSE@UDI

- Subdue is a main memory algorithm.
- Good performance for small data sizes
- Entire graph (or its adjacency matrix) is constructed in main memory before applying the mining algorithm
- Takes a very long time (in hours) to initialize for 1600K edges and 800K vertices graph
- Scalability is an issue
 - Improve performance, if possible
 - Scale the algorithm to data set of any size

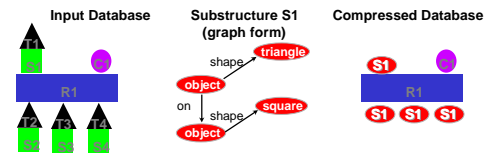


PAKDD'06 Tutorial: SC
4/11/2006

Slide 101

Example

ITLAB
CSE@UDI



PAKDD'06 Tutorial: SC
4/11/2006

Slide 102

MDL Principle

ITLAB
CSE@UDI

- Theory to minimize description length (DL) of data; information theoretic approach
- Has been shown to be good across domains
- Evaluates substructures based on their ability to compress DL of graph
- Description length = $DL(S) + DL(G/S)$
 - Depends upon the representation
 - Substructure that best compresses the original is chosen



PAKDD'06 Tutorial: SC
4/11/2006

Slide 103

MDL Principle (cont.)

ITLAB
CSE@UDI

- Minimizes description length (DL) of data
- Substructures are evaluated based on their ability to compress the DL of the entire graph
- $MDL = \text{description length of the compressed graph} / \text{description length of the original graph}$

$$MDL = \frac{DL(G)}{DL(S) + DL(G|S)}$$

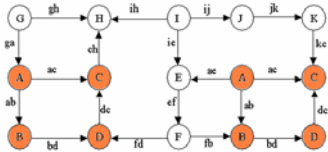
- $DL(G)$ – Description length of the input graph
- $DL(S)$ – Description length of sub graph
- $DL(G|S)$ – Description length of the graph given the sub graph



PAKDD'06 Tutorial: SC
4/11/2006

Slide 104

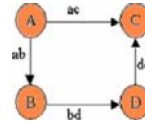
Example: Subdue



Input

- The input is a file, with all the vertex labels, vertex numbers, edges (using vertex numbers) and the edge directions

```
v 1 A
v 2 B
v 3 C
v 4 D
d 1 2 ab
d 1 3 ac
d 2 4 bd
d 4 3 dc
```



- 'd' stands for a directed edge and 'u' stands for undirected. 'e' stands for directed unless specified as -undirected at the command prompt.



Subdue Approach

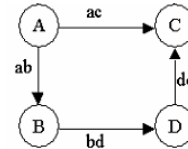
- Create a substructure for each unique vertex label
- Expand each substructure by adding an edge
- Maintain **beam** number of substructures for expansion
- Halting conditions
 - Discovered substructures > **limit**
 - List maintaining the substructures to be expanded becomes empty
 - Max size** of substructure to be discovered is reached



Output

- Output
Substructure: value = 1.21789, instances = 2
Graph (4v,4e):

```
v 1 A
v 2 C
v 3 B
v 4 D
d 1 2 ac
d 1 3 ab
d 2 4 dc
d 4 3 bd
```



Overview (Contd.)

Input Graph

1-edge instances

PAKDD'06 Tutorial: SC
4/11/2006

Slide 109

1 edge pruning

1 edge instances

Frequent Substructures (count)

Substructures After pruning

Instances of Un-pruned substructures retained

Group by

PAKDD'06 Tutorial: SC
4/11/2006

Slide 110

Generating 2 edge substructures

1 edge instances

1-edge instances

2-edge instances

Group by

Frequent 2-edge Substructures (count)

Instances of frequent substructures

Slide 111

PAKDD'06 Tutorial: SC
4/11/2006

Slide 111

Representing input graph using relations

Input stored in two tables

Vertices

vertexno	vertexlabel
1	A

Edges

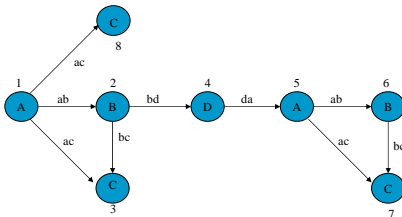
vertex1	vertex2	edgename
1	2	ab

PAKDD'06 Tutorial: SC
4/11/2006

Slide 112

Representation of a Substructure

FTL/AB (U) CSE/UD/1



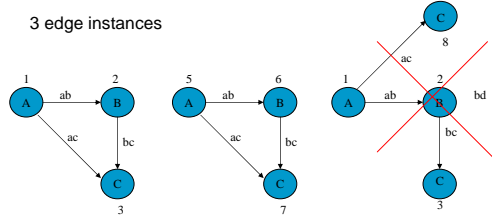
PAKDD'06 Tutorial: SC
4/11/2006

Slide 113

Representation (Contd.)

FTL/AB (U) CSE/UD/1

3 edge instances



Isomorphic instance Count 2

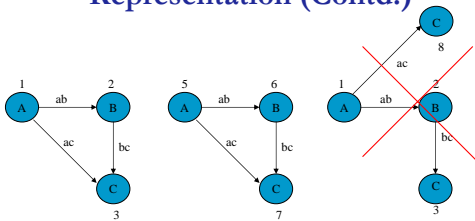


PAKDD'06 Tutorial: SC
4/11/2006

Slide 114

Representation (Contd.)

FTL/AB (U) CSE/UD/1



RDB instance_3 (instances of size 3)

V1	V2	V1V3	V2V4	V1L	V2L	V3L	V4L	E1	E2	E3	F1	T1	F2	T2	F3	T3
1	2	1	3	2	0	3	A	3	B	A	C	B	-	C	ab	C
5	6	5	7	6	0	7	A	7	B	A	C	B	-	C	ab	C
1	2	1	3	2	8	3	A	8	B	A	C	B	-	C	ab	C

Count 2

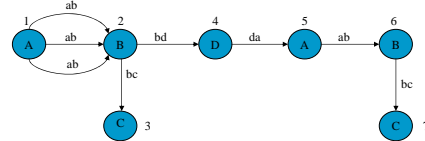


PAKDD'06 Tutorial: SC
4/11/2006

Slide 115

Need for Edge numbers

FTL/AB (U) CSE/UD/1



vertex1	vertex2	edge1	vertex1name	vertex2name
1	2	ab	A	B
1	2	ab	A	B
1	2	ab	A	B
-	-	-	-	-

EDB-Oneedge table

vertex1	vertex2	edge No	edge1	vertex1name	vertex2name
1	2	1	ab	A	B
1	2	2	ab	A	B
1	2	3	ab	A	B
-	-	-	-	-	-

HDB-Oneedge table

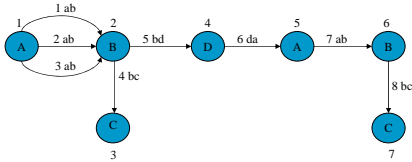


PAKDD'06 Tutorial: SC
4/11/2006

Slide 116

Constrained Expansion

FTL/AB
CSE/UDI



```
INSERT INTO instance_n
( SELECT i.vertex1 .. i.vertex(n), o.vertex2, i.vertex1name .. i.vertex(n)name,
  o.vertex2name, i.edge1 .. i.edge(n), o.edge, i.ext1 , 1
FROM instance(n-1) i, onedge o
WHERE i.vertex(k) = o.vertex1 and i.vertex(k+1) < o.vertex2 and .. i.vertex(n) <
o.vertex2
)
```



PAKDD'06 Tutorial: SC
4/11/2006

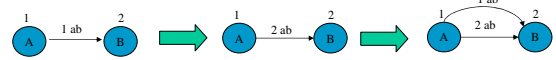
Slide 117

Unconstrained expansion

FTL/AB
CSE/UDI

Instance_1 table

HDB-Oneedge table



vertex1	vertex2	edgeNo	edge1	vertex1name	vertex2name
1	2	1	ab	A	B
.

Instance_1 table

=

vertex1	vertex2	edgeNo	edge	vertex1name	vertex2name
1	2	2	ab	A	B
.

HDB-Oneedge table

WHERE i.vertex1 = o.vertex1 and o.edgeNo <> i.edge1No



PAKDD'06 Tutorial: SC
4/11/2006

Slide 118

Expansion

FTL/AB
CSE/UDI

- To expand a 1-edge substructure with 2 vertices V1 and V2, we need to do:
 - Self edge substructures on v1 and v2
 - Multiple edges between v1 and v2
 - Outgoing edges from v1 and v2
 - Incoming edges from v1 and v2
- In general, to expand a substructure of size n, we need $n^2 + 2n$ queries

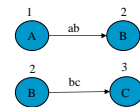
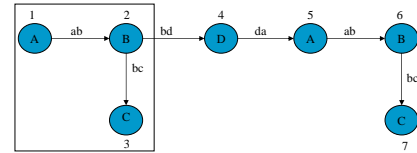


PAKDD'06 Tutorial: SC
4/11/2006

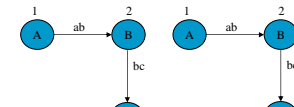
Slide 119

Unconstrained expansion & Pseudo-duplicates

FTL/AB
CSE/UDI



One edge instances



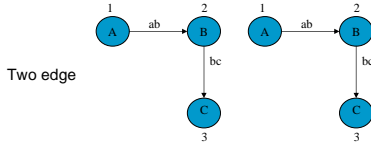
Two edge instances



PAKDD'06 Tutorial: SC
4/11/2006

Slide 120

Pseudo duplicates (Contd..)



Instance_2 table

V1	V2	V3	V1L	V2L	V3L	E1	E2	F1	T1	F2	T2
1	2	3	A	B	C	AB	BC	1	2	2	3
2	3	1	B	C	A	BC	AB	2	3	3	1
.



Pseudo duplicates (Contd..)

In SQL, only rows can be sorted

Tasks:

- Convert **columns into rows**
- **Sort the rows**
- Convert **rows into columns**

Instance_2 table

V1	V2	V3	V1L	V2L	V3L	E1	E2	F1	T1	F2	T2
2	3	1	B	C	A	BC	AB	1	2	3	1
.

Unsorted_V

V	VL	POS
2	B	1
3	C	2
1	A	3
.	.	.

Unsorted_E

E	F	T
BC	1	2
AB	3	1
.	.	.



Updating connectivity attributes

Unsorted_V

V	VL	POS
2	B	1
3	C	2
1	A	3
.	.	.

Sorted_V

V	VL	POS	NEW POS
1	A	3	1
2	B	1	2
3	C	2	3
.	.	.	.

Sorted_V

V	VL	POS	NEW POS
1	A	3	1
2	B	1	2
3	C	2	3
.	.	.	.

Unsorted_E

E	F	T
BC	1	2
AB	3	1
.	.	.

Sorted_V

V	VL	POS	NEW POS
1	A	3	1
2	B	1	2
3	C	2	3
.	.	.	.

Updated_E

E	F	T
BC	2	3
AB	1	2
.	.	.

Sorted_E

E	F	T
BC	2	3
.	.	.

Sort on F, T



Reconstructing instance table

Sorted_V

V	VL	POS	NEW POS
1	A	3	1
2	B	1	2
3	C	2	3
.	.	.	.

Sorted_V

V	VL	POS	NEW POS
1	A	3	1
2	B	1	2
3	C	2	3
.	.	.	.

Sorted_V

V	VL	POS	NEW POS
1	A	3	1
2	B	1	2
3	C	2	3
.	.	.	.

Sorted_E

E	F	T
AB	1	2
BC	2	3
.	.	.

Sorted_E

E	F	T
AB	1	2
BC	2	3
.	.	.

2n+1 Way Join for reconstruction

Instance_2 table

V1	V2	V3	V1L	V2L	V3L	E1	E2	F1	T1	F2	T2
1	2	3	A	B	C	AB	BC	1	2	2	3
.



Pseudo duplicates (Contd..)

FTL/AB
(U)
CSE/UD/1

- Group By Vertex numbers and edge direction attributes
- Retain one and eliminate other

V1	V2	V3	V1L	V2L	V3L	E1	E2	F1	T1	F2	T2
1	2	3	A	B	C	AB	BC	1	2	2	3
1	2	3	A	B	C	AB	BC	1	2	2	3
.

Instance_2 table

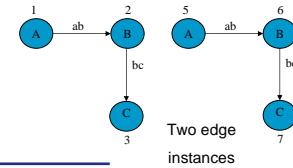
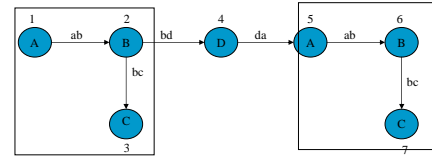


PAKDD'06 Tutorial: SC
4/11/2006

Slide 125

Canonical label ordering

FTL/AB
(U)
CSE/UD/1



PAKDD'06 Tutorial: SC
4/11/2006

Slide 126

Canonical label ordering

FTL/AB
(U)
CSE/UD/1

- These two are NOT duplicates
- They need to be recognized as isomorphic to each other
- In order to do this vertices and labels need to be canonically ordered
- This process for canonical ordering is similar to the process used for pseudo duplicate elimination

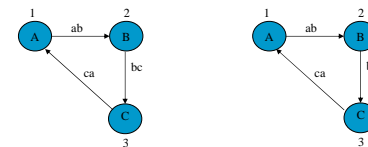
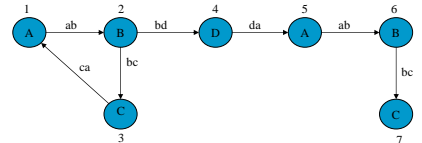


PAKDD'06 Tutorial: SC
4/11/2006

Slide 127

Cycles – Marking repeated vertex

FTL/AB
(U)
CSE/UD/1



Three edge starting with ab

Three edge starting with bc

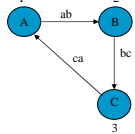


PAKDD'06 Tutorial: SC
4/11/2006

Slide 128

Cycles – Marking repeated vertex (Contd..)

FTL/AB
(3)
CSE/UDI



V1	V2	V3	V4	V1L	V2L	V3L	V4L	E1	E2	E3	F1	T1	F2	T2	F3	T3
1	2	3	1	A	B	C	A	AB	BC	CA	1	2	2	3	3	4
2	3	1	2	B	C	A	B	BC	CA	AB	1	2	2	3	3	4

Instance_3 table

V1	V2	V3	V4	V1L	V2L	V3L	V4L	E1	E2	E3	F1	T1	F2	T2	F3	T3
0	2	3	0	A	B	B	-	AB	BC	CA	2	3	3	3	3	2
0	3	2	0	B	C	A	-	BC	CA	AB	2	3	2	3	3	2

Instance_3 table with vertex invariants 0's and -'s

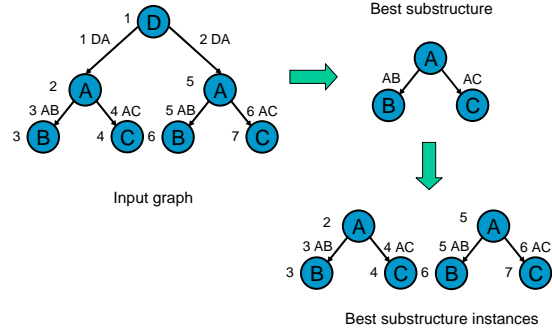


PAKDD'06 Tutorial: SC
4/11/2006

Slide 129

Hierarchical Reduction

FTL/AB
(3)
CSE/UDI

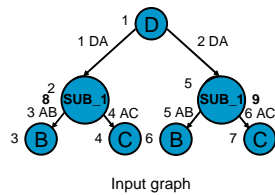
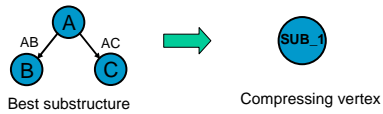


PAKDD'06 Tutorial: SC
4/11/2006

Slide 130

Hierarchical Reduction (Contd..)

FTL/AB
(3)
CSE/UDI



PAKDD'06 Tutorial: SC
4/11/2006

Slide 131

Hierarchical Reduction (Contd..)

FTL/AB
(3)
CSE/UDI

Tasks in Hierarchical Reduction:

1. Select the best substructure after each iteration
2. Identify the instances of the best substructure
3. For each instance
 - (1) Remove the vertices from the input graph
 - (2) For every instance removed include a new vertex to the input graph
 - (3) Remove the edges from the input graph
 - (4) Update the vertex number of the edges that are incident on or going out of the compressed instance
4. The compressed input graph participates in the next iteration

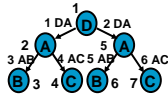


PAKDD'06 Tutorial: SC
4/11/2006

Slide 132

Selecting the best substructure

FTL/AB
CSE/UDI



subfold_2

V1L	V2L	V3L	E1	E2	F1	T1	F2	T2	COUNT	DMDL
A	B	C	AB	AC	1	2	1	3	2	1.71
D	A	B	DA	AB	1	2	2	3	2	1.34
D	A	C	DA	AC	1	2	2	3	2	1.34
D	A	A	DA	DA	1	2	1	3	1	0.75

Best substructure is one with highest DMDL value

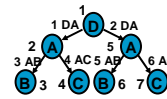


PAKDD'06 Tutorial: SC
4/11/2006

Slide 133

Identify instances of best substructure

FTL/AB
CSE/UDI



V1	V2	V3	V1L	V2L	V3L	E1N	E2N	E1	E2	F1	T1	F2	T2	DMDL
2	3	4	A	B	C	3	4	AB	AC	1	2	1	3	1.71
5	6	7	A	B	C	5	6	AB	AC	1	2	1	3	1.71

BestInstances

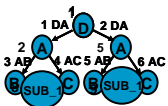


PAKDD'06 Tutorial: SC
4/11/2006

Slide 134

Updating vertex table

FTL/AB
CSE/UDI



V1	V2	V3	V1L	V2L	V3L	E1N	E2N	E1	E2	F1	T1	F2	T2	DMDL
2	3	4	A	B	C	3	4	AB	AC	1	2	1	3	1.71
5	6	7	A	B	C	5	6	AB	AC	1	2	1	3	1.71

BestInstances

Vertex table

V	VL	VL
1	D	D
2	A	SUB_1
3	B	SUB_1
4	C	
5	A	
6	B	
7	C	

```
DELETE FROM VertexTable
WHERE VertexNo IN (
  (SELECT V1 FROM BestInstances)
  UNION (SELECT V2 FROM BestInstances)
  ...
  UNION (SELECT Vn+1 FROM BestInstances))
```

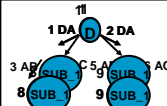


PAKDD'06 Tutorial: SC
4/11/2006

Slide 135

Updating oneedge table

FTL/AB
CSE/UDI



V1	V2	V3	V1L	V2L	V3L	E1N	E2N	E1	E2	F1	T1	F2	T2	DMDL
2	3	4	A	B	C	3	4	AB	AC	1	2	1	3	1.71
5	6	7	A	B	C	5	6	AB	AC	1	2	1	3	1.71

BestInstances

Oneedge table

EL	EN	W1L	W2L	V1	V1/2
DA	1	D	D	1	1
DA	2	D	SUB_1	1	4
AB	3	A	B	2	3
AC	4	A	C	2	4
AB	5	A	B	5	6
AC	6	A	C	5	7

```
DELETE FROM oneedge
WHERE EL IN (
  (SELECT E1N FROM BestInstances)
  UNION (SELECT E2N FROM BestInstances)
  ...
  UNION (SELECT E1 FROM BestInstances))
WHERE V2 IN (
  (SELECT V1 FROM BestInstances WHERE rownum = 1)
  ...
  UNION (SELECT Vn+1 FROM BestInstances WHERE rownum = 1))
```



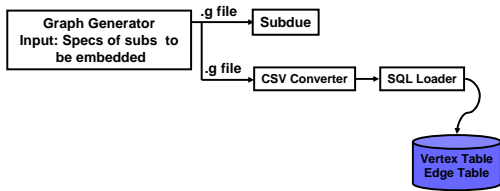
PAKDD'06 Tutorial: SC
4/11/2006

Slide 136

Experimental Results

Setup:

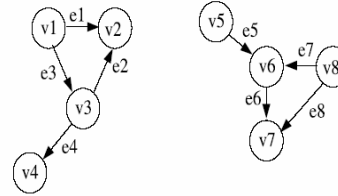
- Input graphs generated using the graph generator developed by AI Lab
- Platform: Linux
- Database: Oracle 10g
- Machine's memory: 2 Gbytes
- Number of processors: 2



PAKDD'06 Tutorial: SC
4/11/2006

Slide 137

No cycles and multiple edges



Dataset	Instances
50V100E	4
250V500E	15
500V1000E	30
1KV2KE	60
2.5KV5KE	150
5KV10KE	300
7.5KV15KE	450
10KV20KE	600
15KV30KE	900
20KV40KE	1200
50KV100KE	3000
100KV200KE	6000
200KV400KE	12000
400KV800KE	24000
800KV1600KE	48000

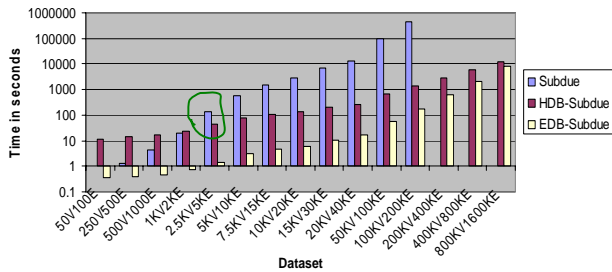


PAKDD'06 Tutorial: SC
4/11/2006

Slide 138

No cycles and multiple edges

Beam 4, MaxSize 5, Iterations 1



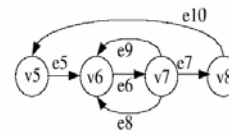
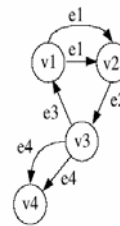
- Running time grows exponentially with graph input size
- Subdue crossover – 2.5KV5KE



PAKDD'06 Tutorial: SC
4/11/2006

Slide 139

With cycles and multiple edges

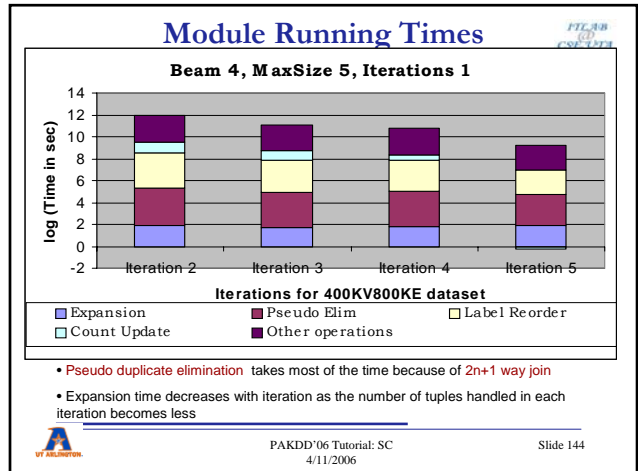
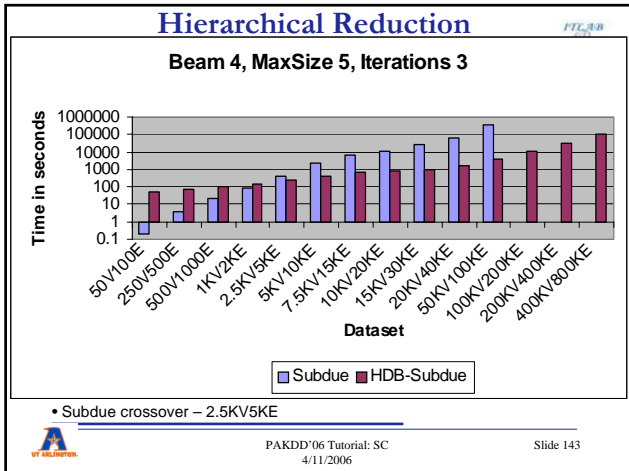
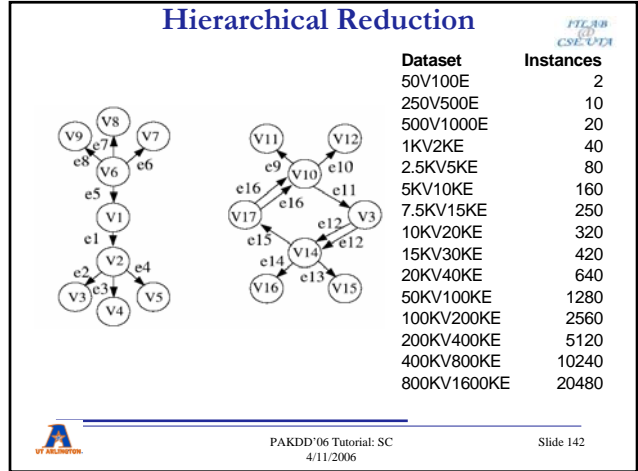
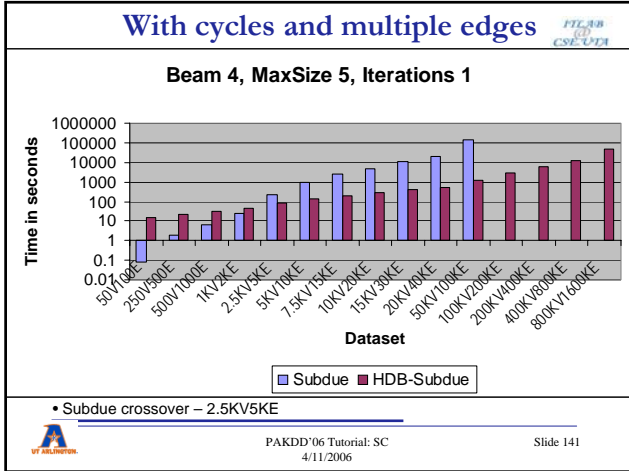


Dataset	Instances
50V100E	4
250V500E	15
500V1000E	30
1KV2KE	60
2.5KV5KE	150
5KV10KE	300
7.5KV15KE	450
10KV20KE	600
15KV30KE	900
20KV40KE	1200
50KV100KE	3000
100KV200KE	6000
200KV400KE	12000
400KV800KE	24000
800KV1600KE	48000



PAKDD'06 Tutorial: SC
4/11/2006

Slide 140



Tutorial Outline

ITL:AB
(U)
CSE:UDI

- Data Mining Overview
- Database Mining
 - Need
 - Spectrum
- Database Mining Architectures
- Mining Using SQL
 - Association Rules
 - Graph Mining
 - *Significant Interval Discovery*
 - *Event Pattern Discovery*
- **Summary and Challenges**
- References



PAKDD'06 Tutorial: SC
4/11/2006

Slide 145

Summary

ITL:AB
(U)
CSE:UDI

- **First generation systems**
 - use single DM technique
 - work with small data sets (heavy use of sampling)
 - where is the beef? Extract symbolic patterns with little knowledge about the data
- **Second generation systems**
 - use multiple DM techniques
 - work with large datasets (loose coupling with DBMS)
 - where is the beef? Apply multiple techniques on the same sample to extract variety of patterns



PAKDD'06 Tutorial: SC
4/11/2006

Slide 146

Summary (contd.)

ITL:AB
(U)
CSE:UDI

- **Third generation systems**
 - use multiple DM techniques, automatically selecting the most appropriate data, technique(s) and result(s)
 - work with very large data sets (tight connection with DW, embedded into DBMS)
 - where is the beef? full exploitation of parallelism



PAKDD'06 Tutorial: SC
4/11/2006

Slide 147

Challenges

ITL:AB
(U)
CSE:UDI

- Primitive operators inside DBMS
- Optimization of self-joins
- Efficient pseudo duplicate elimination
- Query optimization and plan generation
- Mining-aware DBMSs and SQL-aware mining systems
- Perhaps concurrency control and recovery are not needed and if turned off, can result in better performance



PAKDD'06 Tutorial: SC
4/11/2006

Slide 148

Current work

- Extending to FSG, gSpan
- Use of indexing
- Infer new optimizations that are needed

- Some things may need better procedural support (e.g., UDFs for pseudo duplicate elimination)



Tutorial Outline

- Data Mining Overview
- Database Mining
 - Need
 - Spectrum
- Database Mining Architectures
- Mining Using SQL
 - Association Rules
 - Graph Mining
 - Significant Interval Discovery
 - Event Pattern Discovery
- Summary and Challenges
- References



References

- Thuraisingham, B., *A Primer for Understanding and Applying Data Mining*. IEEE, 2000. Vol. 2, No. 1, p. 28-31.
- Thomas, S., *Architectures and optimizations for integrating Data Mining algorithms with Database Systems*, PhD thesis, CISE department 1998, University of Florida: Gainesville.
- Agrawal, R., T. Imielinski, and A. Swami. *Mining Association Rules between sets of items in large databases*. in *ACM SIGMOD International Conference on the Management of Data*. 1993. Washington, D.C.
- Agrawal, R. and R. Srikant. *Fast Algorithms for mining association rules*. in *20th Int'l Conference on Very Large Databases (VLDB)*. 1994.
- Sarasere, A., E. Omiecinsky, and S. Navathe. *An efficient algorithm for mining association rules in large databases*. in *21st Int'l Cong. on Very Large Databases (VLDB)*. 1995. Zurich, Switzerland.
- Shenoy, P., et al. *Turbo-charging Vertical Mining of Large Databases*. in *ACM SIGMOD Int'l Conference on Management of Data*. 2000. Dallas.
- Han, J., J. Pei, and Y. Yin. *Mining Frequent Patterns without Candidate Generation*. in *ACM SIGMOD Int'l Conference on Management of Data*. 2000. Dallas.
- Houtsma, M. and A. Swami. *Set-Oriented Mining for Association Rules in Relational Databases*. in *11th International Conference on Data Engineering (ICDE)*. 1995.



References

- Han, J., et al. *DMQL: A data mining query language for relational database*. in *ACM SIGMOD workshop on research issues on data mining and knowledge discovery*. 1996. Montreal.
- Meo, R., G. Psaila, and S. Ceri. *A New SQL-like Operator for Mining Association Rules*. in *Proceedings of the 22nd VLDB Conference*. 1996. Mumbai, India.
- Agrawal, R. and K. Shim. *Developing tightly-coupled Data Mining Applications on a Relational Database System*. 1995, IBM Almaden Research Center: San Jose, California.
- Sarawagi, S., S. Thomas, and R. Agrawal. *Integrating Association Rule Mining with Relational Database System: Alternatives and Implications*. in *ACM SIGMOD Int'l Conference on Management of Data*. 1998. Seattle, Washington.
- Duggir, M., *A Layered Optimizer for Mining Association Rules over RDBMS*. in *CSE Department*. 2000, University of Florida: Gainesville.
- Thomas, S. and Chakravarthy, S. Performance evaluation and optimization of join queries for association rule mining. *Proc. of the First International Conference on Data Warehousing and Knowledge Discovery, DaWaK '99*, Florence, Italy, August 1999
- R. Balachandran, S. Padmanabhan, and S. Chakravarthy, "Enhanced DB-Subdue: Supporting Subtle Aspects of Graph Mining Using a Relational Approach", To appear in PAKDD, Singapore, April 2006 (Short paper)
- A. Srinivasan, S. Sreshta, and S. Chakravarthy, "Discovery of Significant Intervals in Sequential Data", in the Proc. of 1st ADBIS Workshop on Data Mining and Knowledge Discovery, Tallinn, Estonia, September 2005, pp. 87-98.



References

- H. Kona, S. Chakravarthy, and A. Arora, "SQL-Based Approach to Incremental Association Rule Mining", in the Proc. of 1st ADBIS Workshop on Data Mining and Knowledge Discovery, Tallinn, Estonia, September 2005, pp. 11-24
- Sharma Chakravarthy, Ramji Beera, and Ram Balachandran, DB-Subdue: Database Approach to Graph Mining, In PAKDD conference, Sydney, May 2004.
- P. Mishra and S. Chakravarthy, "Performance Evaluation of SQL-OR Variants for Association Rule Mining", in Proc. Of DaWaK (Data warehousing and Knowledge Discovery), September 2003, Prague.
- P. Mishra and S. Chakravarthy, "Performance Evaluation and Analysis of K-way join variants for Association Rule Mining", in Proceedings of BNCOD 2003, Sheffield, UK, July 2003, 95-114.
- M. Dudgikar, S. Chakravarthy, R. Luzzi, and L. Wong, "A Layered Optimizer for Mining Association Rules over Relational Database Management Systems", 2003 International Conference on Artificial Intelligence (IC-AI2003), June 2002, Monte Carlo Resort, Las Vegas, Nevada
- S. Chakravarthy and H. Zhang, Visualization of association Rules over RDBMSs, in the proceedings of ACM SAC 2003, Melbourne, FL, March 2003 (Multi-media and Visualization Track).
- P. Mishra and S. Chakravarthy, "Performance Evaluation and Analysis of K-way join variants for Association Rule Mining", in Proceedings of BNCOD 2003, Sheffield, UK, July 2003, 95-114.
- Mr. Srihari Padmanabhan, "Relational Database Approach to Graph Mining and Hierarchical Reduction", Fall 2005 <http://itlab.uta.edu/itlabweb/students/shr05ms.pdf>
- Mr. Sunit Sreshta, "SQL_Based Approach to Significant Interval Discovery in Time-Series Data", Summer 2005 <http://itlab.uta.edu/itlabweb/students/shr05ms.pdf>



References

- R. Balachandran, "Relational Approach to Modeling and Implementing Subtle Aspects of Graph Mining", Fall 2003. <http://www.cse.uta.edu/Research/Publications/Downloads/CSE-2003-41.pdf>
- Ms. A. Krishnamurthy, "Significant Interval and Episode Discovery in Time-Series Data", Fall 2003. <http://www.cse.uta.edu/Research/Publications/Downloads/CSE-2003-39.pdf>
- Mr. P. Mishra, "Performance Evaluation and Analysis of SQL-Based Approaches for Association Rule Mining", Fall 2002. <http://www.cse.uta.edu/Research/Publications/Downloads/CSE-2003-3.pdf>
- Mr. Hongen Zhang, "Mining and Visualization of Association Rules in Relations DBMSs", Summer 2000. <http://itlab.uta.edu/sharma/People/ThesisWeb/etd.pdf>
- S. Thomas and S. Chakravarthy, "Incremental Mining of Constrained Associations", in Proc. of High Performance Computing (HiPC), Bangalore, India, Dec. 2000.

