

Seamless Integration of DM with DBMS and Applications

Hongjun Lu

Hong Kong University of Science & Technology



Email: *luhj@cs.ust.hk*

URL: *http://www.cs.ust.hk/~luhj*



-
-
-

Main Arguments

It is a critical issue

to sustain KDD research.

to make DM a main stream technology

It is not really a new problem, but worth more attention from researchers

It should be addressed from various directions

•
•
•

A Very Hot Research Topic Could Die

Deductive database

Excellent topic/work

Little application support

- Absorbed by RDBMS Extensions in RDBMS

OODB

Good piece of work

There are application support

Did not become main stream technology

- Absorbed by RDBMS OR-DBMS

-
-
-

What Makes KDD Different?

Experience of RDBMS

Turn theory a popular technology

What will really distinguish KDD from other data analysis research work

Scalability?

Making sophisticate /theoretical work a main stream technology on the market!

Making DM a Main Stream Technology

Data is everywhere, database is every where, why not data mining?

Unfortunately, not yet

DM has been then for a decade, not a main stream technology yet

Perceived to be sophisticated technology, usable only by specialists

Long, expensive projects

Stand-alone, loosely-coupled with data infrastructures

Difficult to infuse into existing mission-critical applications

-- R. Agrawal, KDD'99

Efforts Have Been Made

Has been an open problem

Emphasized in tutorials, speeches

Various efforts have been made

Vendors:

- both DM software vendors/ DBMS vendors

Researchers:

- DBMS side: functionality/language extensions
- DM side: DBMS/SQL-aware mining

-
-
-

Association Rule Mining and DBMS

Houtsma and Swami, ICDE95

SQL implementation of Apriori-based association rule mining

Agrawal and Shim, KDD96

Tight coupling, UDF approach

Sarawagi, Thomas and Agrawal, SIGMOD98

Association rule mining based on SQL-92 and ORDBMS

Thomas and Sarawagi, KDD98

Generalized AR/sequential mining using SQL queries

Thomas and Chakravarthy, DaWak99

Performance evaluation of SQL-92 based AR mining

Pramudiono, Shintani, Tamura and Kitsuregawa, PAKDD99,

Yoshizawa, Pramudiono, and Kitsuregawa, DaWak00

Parallel mining of generalized AR using SQL

-
-
-

SQL-Aware Classification

Wang, Iyer, and Vitter, *IDEAL98*

Decision tree based algorithm

Using UDF to implement required computations

Chaduri, Fayyad & Bernhardt, *ICDE'99*

Middleware between SQL Server & classification algorithms

Emphasizing performance

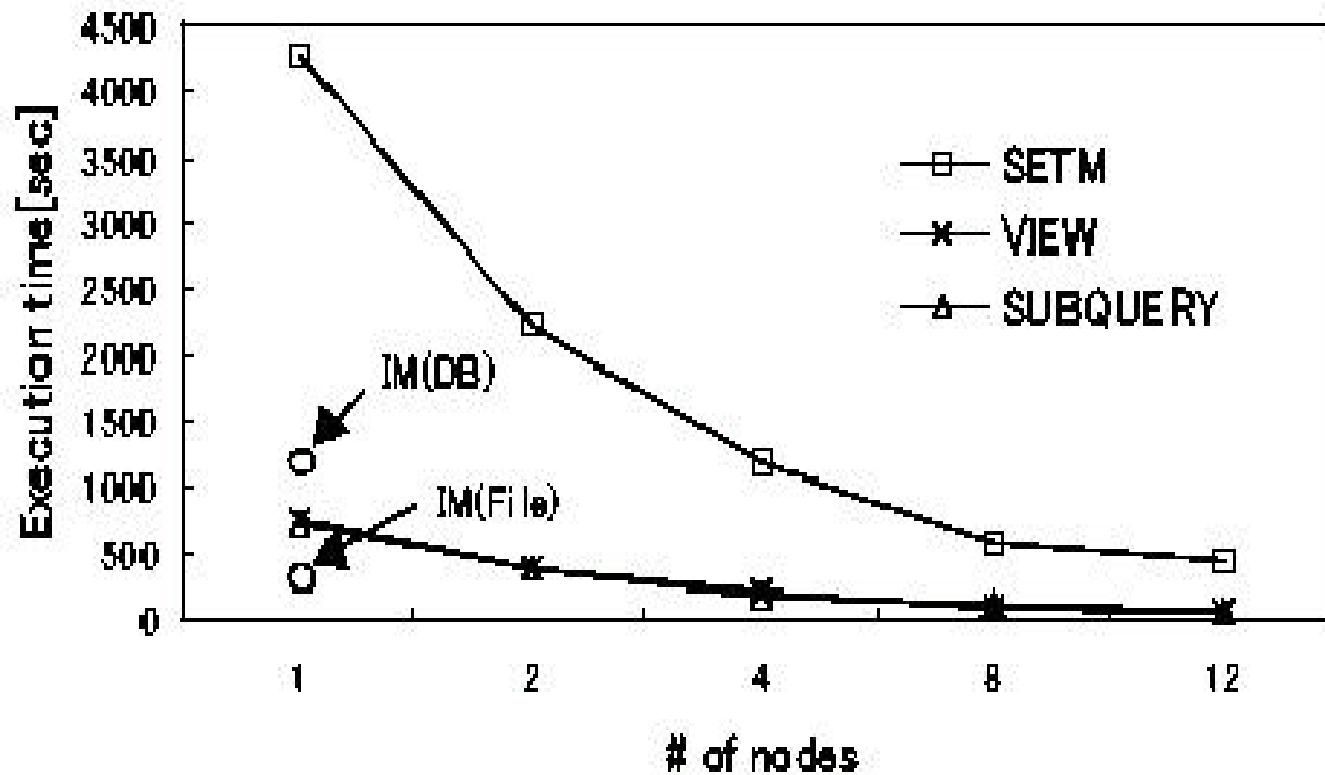
Lu and Liu, *VLDB2000*

Scalable classifier

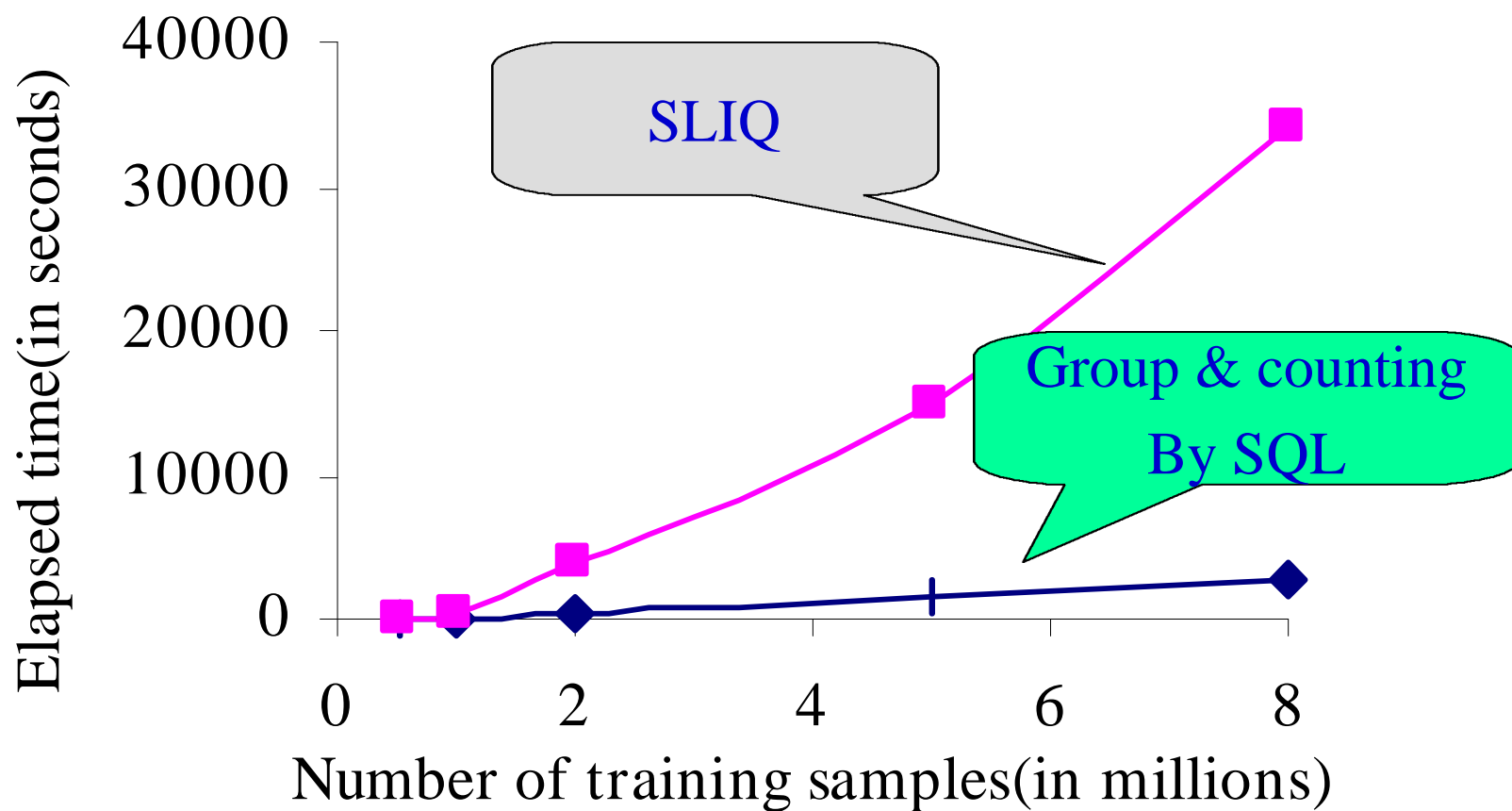
Implemented using DB2 SQL statements

How about Performance

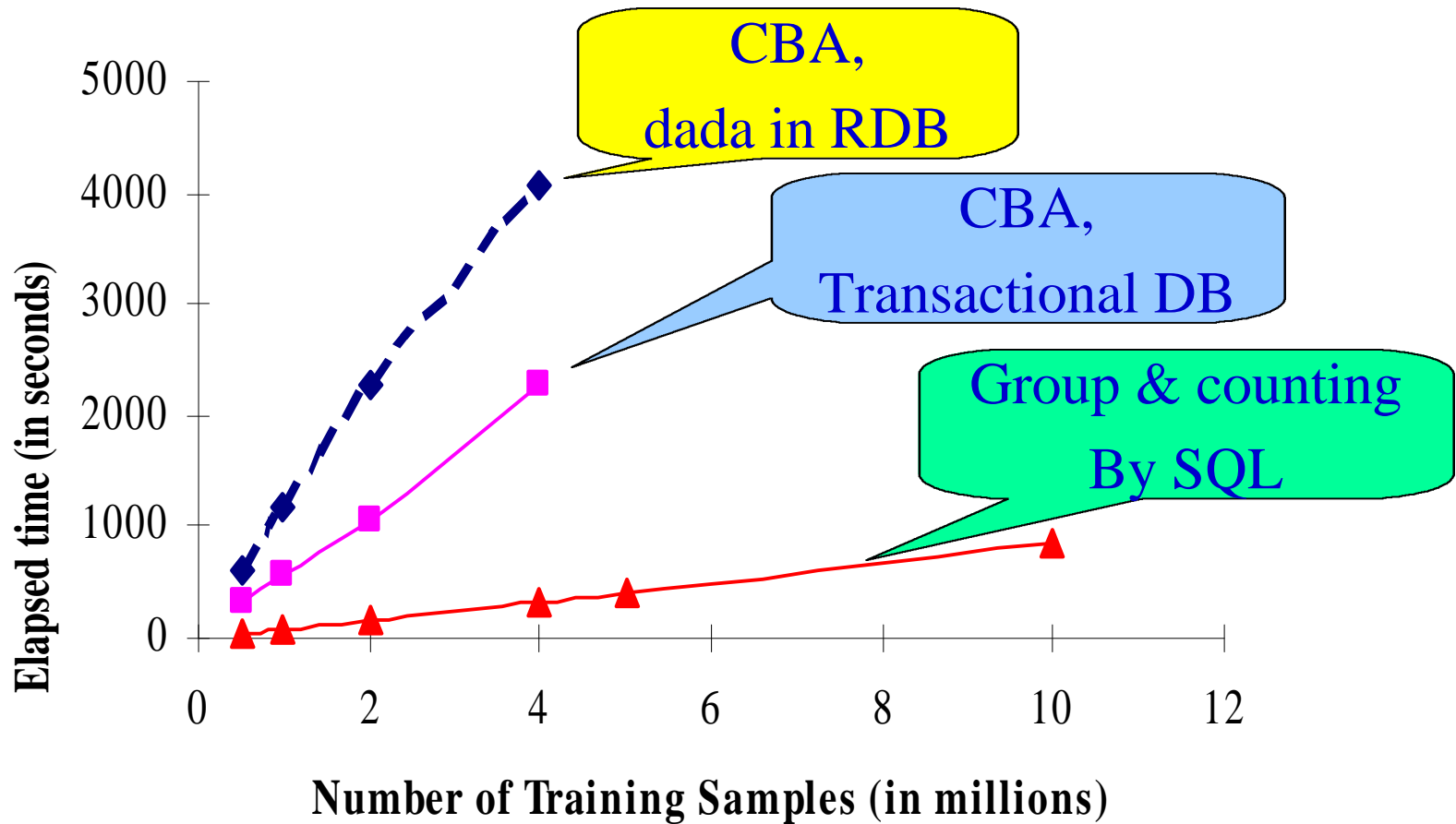
Yoshizawa, Pramudiono, Kitsuregawa DaWak'00
IBM RS6000 SP (12 nodes)



Compare with SLIQ (Function 5)



Compare with CBA (Function 10)



-
-
-

Seamless Integration

A wide range of approaches

Integration of DM & DBMS

ODBC data access

Elegant data staging/caching

Stored procedures and user defined functions

DM as DBMS applications

Integration DM & applications

Uniform/standard interface

Application-friendliness data mining

tightness



-
-
-

Integration of DM and DBMS

machine learning/statistical algorithms

main memory based (e.g. C4.5)

disk aware algorithms

data on the disk, I/O cost should be minimized (Agrawl'92)

DBMS aware algorithms

tight/smart control of interaction between DM algorithms and DBMS (e.g. Chaduri, Fayyad & Bernhardt, ICDE'99)

SQL aware algorithms

employ stored procedures/UDF (e.g. Wang, Iyer, and Vitter, *IDEAL98*)

DM implemented as embedded SQL programs

employ DBMS functions only (e.g. Lu & Liu, VLDB00)

Our Experience -- Classification

Starting with efficient classification

one of most popular DM problems

Using RDBMS native functions

push as much data processing to RDBMS as possible

- Performance
- Scalability
- Leverage

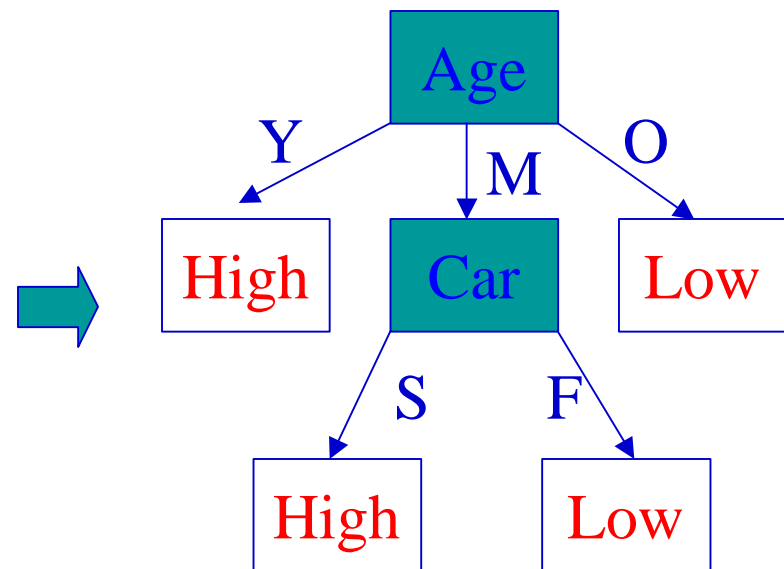
Model (the classifier built) as first class citizen (table) of database

facilitate integration with applications

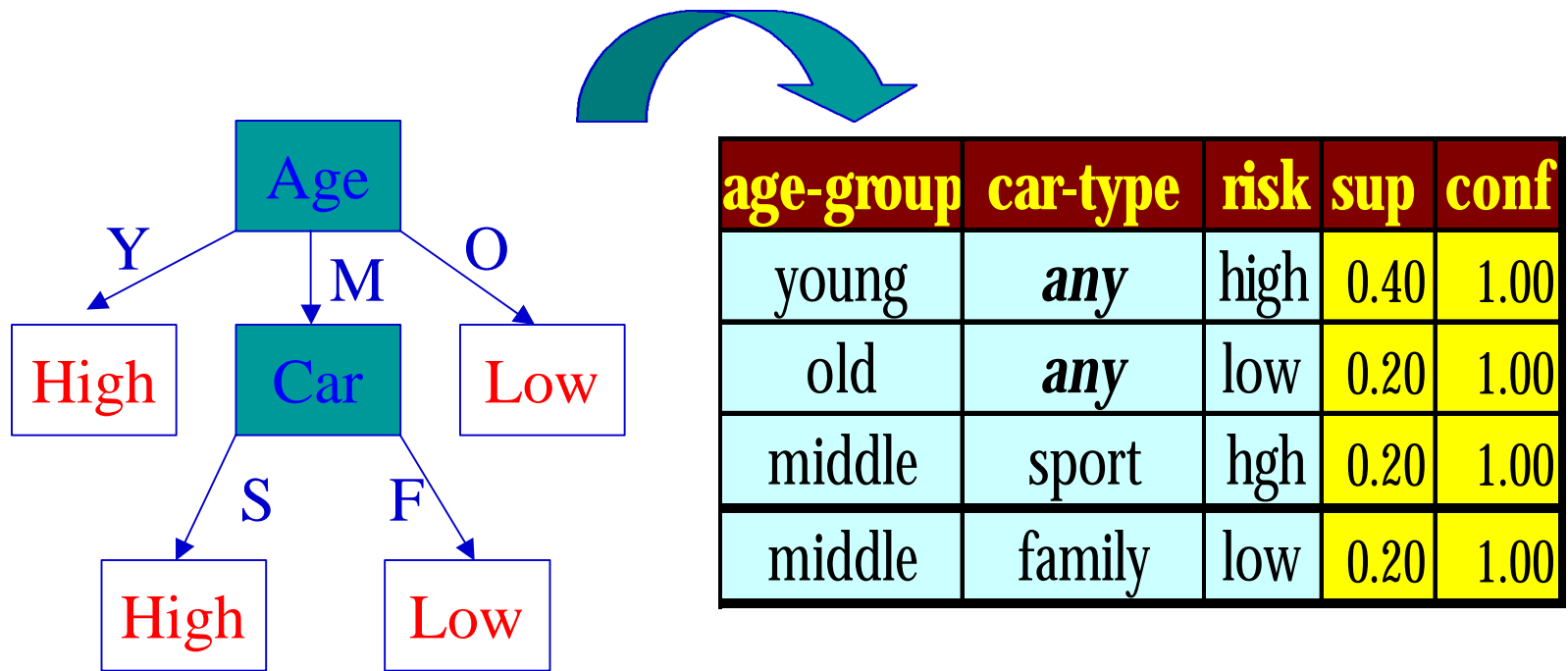
Classification – Decision Trees

Various algorithms have been developed
Decision tree: one of the popular methods

age-group	car-type	risk
young	family	high
young	sport	high
middle	sport	high
old	family	low
middle	family	low



Classifier as Relational Tables



Using the Classifier: Case One

Given tuple $u (a_{1u}, a_{2u}, \dots, a_{nu})$, **one matching row**
 $r_i (a_{1i}, a_{2i}, \dots, a_{ni}, c_i, sup_i, conf_i)$ **found**

Class of u is c_i .

(young, family) ?

risk = high

(middle, family) ?

risk = low

age-group	car-type	risk	sup	conf
young	any	high	0.4	1.00
old	any	low	0.2	1.00
any	sport	high	0.4	1.00
middle	family	low	0.2	1.00

Using the Classifier: Case One

Given tuple $u (a_{1u}, a_{2u}, \dots, a_{nu})$, more than one matching row is found:

select the one with highest confidence

If there are ties, the class with highest support will be assigned to c_u .

If there are still ties, one randomly picked from them will be assigned to c_u .

(old, sport) ?

risk = high

age-group	car-type	risk	sup	conf
young	<i>any</i>	high	0.4	1.00
old	<i>any</i>	low	0.2	1.00
<i>any</i>	sport	high	0.4	1.00
middle	family	low	0.2	1.00

-
-
-

Using the Classifier: Case Three

Given tuple $u (a_{1u}, a_{2u}, \dots, a_{nu})$, no matching row is found Compute $P(C_k | u)$ based on Bayes theorem

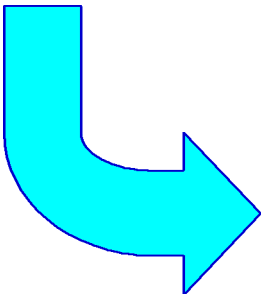
$$P(C_k | u) \propto p(C_k) \prod_{i=1}^n p(a_{iu} | C_k) = \frac{\prod_{i=1}^n p(a_{iu} \wedge C_k)}{p(C_k)^{n-1}}$$

Class C_k with the highest probability is assigned to u

age-group	car-type	risk
young	family	high
young	sport	high
middle	sport	high
old	family	low
middle	family	low

occurrences of samples with the same attribute values

age-group	car-type	risk	count
young	<i>any</i>	high	2
middle	<i>any</i>	high	1
middle	<i>any</i>	low	1
old	<i>any</i>	low	1
<i>any</i>	family	high	1
<i>any</i>	family	low	2
<i>any</i>	sport	high	2
young	family	high	1
young	sport	high	1
middle	sport	high	1
middle	family	low	1
old	family	low	1



GAC:
Grouping and Counting

age-group	car-type	risk	count	conf
young	any	high	2	1.00
middle	any	high	1	0.50
middle	any	low	1	0.50
old	any	low	1	1.00
any	family	high	1	0.33
any	family	low	2	0.67
any	sport	high	2	1.00
young	family	high	1	1.00
young	sport	high	1	1.00
middle	sport	high	1	1.00
middle	family	low	1	1.00
old	family	low	1	1.00



Low confidence

age-group	car-type	risk	sup	conf
young	any	high	0.4	1.00
old	any	low	0.2	1.00
ANY	sport	high	0.4	1.00
middle	family	low	0.2	1.00

Redundant

$Sup = count / N$
 N : # of total samples

$$Conf = \frac{\# \text{ of samples with same attribute values (inclusive of class)}}{\# \text{ of samples with same attribute values (exclusive of class)}}$$


Example of grouping sets query

SELECT age-group, car-type,
risk, count(*) as count

FROM insurance

GROUP BY GROUPING SETS

(age-group, car-type), risk



age-group	car-type	risk
young	family	high
young	sport	high
middle	sport	high
old	family	low
middle	family	low



age-group	car-type	risk	count
middle	-	high	1
middle	-	low	1
old	-	low	1
young	-	high	2
-	family	high	1
-	family	low	2
-	sport	high	2

Computing Class Populations

```

SELECT A1, A2, ..., An, class, count(*)
FROM TrainD
GROUP BY Ak, CUBE (A1, A2, ...,
    Ak-1, Ak+1, ..., An), class
ORDER BY A1, A2, ..., An

```

where A_k is the best splitting attribute.

age-group	car-type	risk
young	family	high
young	sport	high
middle	sport	high
old	family	low
middle	family	low



```

SELECT age-group, car-type, risk, count(*)
FROM insurance
GROUP BY age-group, CUBE (car-type), risk

```

age-group	car-type	risk	count
middle	any	high	1
middle	any	low	1
old	any	low	1
young	any	high	2
middle	sport	high	1
middle	family	low	1
old	family	low	1
young	family	high	1
young	sport	high	1

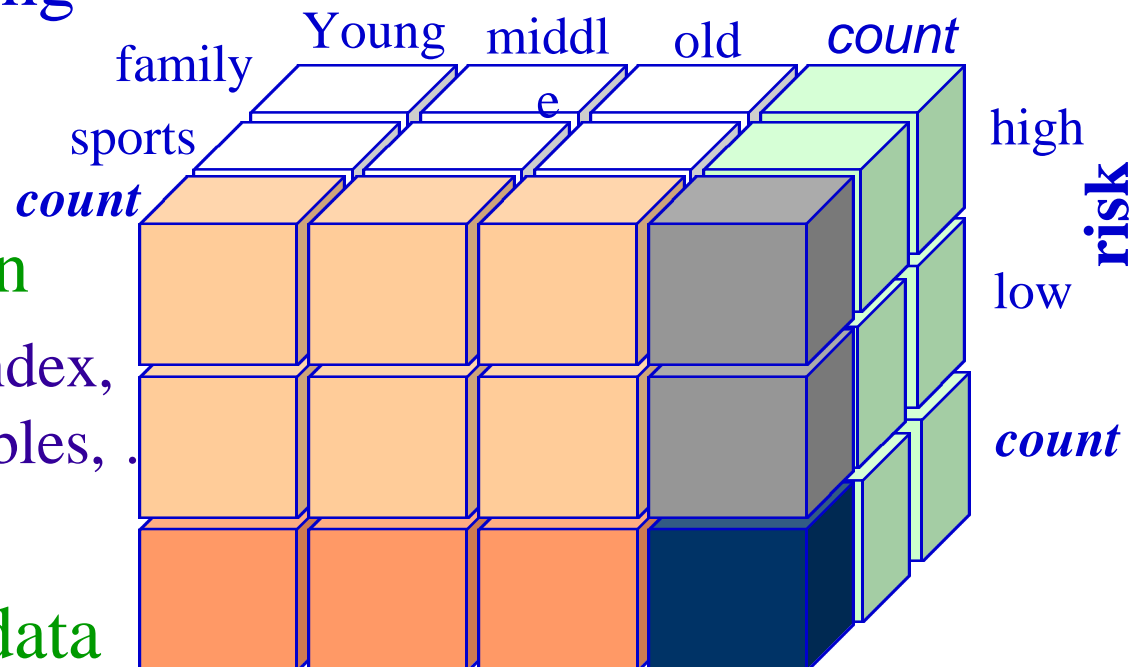
Classification Meets DBMS

The most demanding computation in classification

Class population

- entropy, gini index, contingency tables, .

No more than computing the data cube!



Implementation Issues/Solutions

Data cube computation is expensive in DBMS

Large # of attributes High dimension

Large # of distinct attribute values large # of groups

Optimization techniques addressing the issue

Reducing dimension by one

- selecting a starting attribute

Generating the classifier in iterations:

- each iteration computes a sub-cubes and generates rules involving selected attributes.

-
-
-

What Learned

A SQL based classification algorithm can be developed
fully implemented as an embedded SQL program without
UDF
Classifier obtained is a relational table

Performance seems not an issue

Better classification accuracy

Execution speed & scalability issue solvable.

How about other DM problems?

Integration of DM and Application

Standard interfaces, specifications for

Input data

Mining parameter

Output of mining (the model)

Sample efforts

OLE-DB for data mining

PMML (Predictive Model Markup Language)

- an XML-based to define predictive models for sharing

-
-
-

The Real Challenge

The real challenge faced by users of DM technology/applications

What to mine?

What technique to use?

Given technique, what method/algorithm to use

With chosen method/algorithm, how to set parameters

-
-
-

Application Friendly Mining Algorithms

Algorithm selection

Parameter reduction

Parameter selection/transformation

Algorithm Selection

Auto, semi-auto algorithm selection?

Impossible?

- Set a modest goal
- How about allow you to scan data once/sample data before making decision

Application dependency?

- vertical market
- Turn-key systems
- inadequate research in integrating domain knowledge!

Benchmarking/analyzing existing algorithms

Bench marking data

Data set	Properties				Accuracy					
	#attrs	#classes	# train	# test	C4.5	NB	TAN	CBA	LB	GAC
Australian	14	2	690	CV-10	0.843	0.857	0.852	0.855	0.857	0.883
Chess	36	2	2,130	1,065	0.995	0.872	0.921	0.981	0.902	0.944
Diabetes	8	2	768	CV-10	0.717	0.751	0.765	0.729	0.767	0.767
Flare	10	2	1,066	CV-10	0.812	0.795	0.826	0.831	0.815	0.843
German	20	2	1,000	CV-10	0.717	0.741	0.727	0.732	0.748	0.768
Heart	13	2	270	CV-10	0.767	0.822	0.833	0.819	0.822	0.838
Letter	16	26	15,000	500	0.777	0.749	0.857	0.518	0.764	0.800
Lymph	18	4	148	CV-10	0.784	0.819	0.838	0.773	0.846	0.839
Pima	8	2	768	CV-10	0.711	0.759	0.758	0.730	0.758	0.780
Satimage	36	6	4,435	2,000	0.852	0.818	0.872	0.849	0.839	0.847
Segment	19	7	1,540	770	0.958	0.918	0.935	0.935	0.942	0.943
Splice	60	3	2,126	1,064	0.933	0.946	0.946	0.700	0.946	0.956
Shuttle-small	9	7	38,661	934	0.995	0.987	0.996	0.995	0.994	0.998
Vehicle	18	4	846	CV-10	0.698	0.611	0.709	0.688	0.688	0.681
Voting Records	16	2	435	CV-10	0.957	0.903	0.933	0.935	0.947	0.956
Waveform-21	21	3	300	4,700	0.704	0.785	0.791	0.753	0.794	0.761
Yeast	8	10	1,484	CV-10	0.557	0.581	0.572	0.551	0.582	0.574
AVERAGE					0.810	0.807	0.831	0.787	0.824	0.834

-
-
-

Parameter Transformation

Algorithm-oriented versus application oriented

Providing the mapping between application parameters and algorithm parameters

-
-
-

Conclusions

Seamless integration is a critical issue

The problem can be addressed from different angles

Defining standard interface for data/model exchange

Developing DBMS/SQL based mining algorithms

Making mining algorithms application friendly

.....

The problems are not easy to solve, not too many people are willing to solve, and that is why we, academic researchers, exist!