

Exploratory Analysis in Cube Space

Raghu Ramakrishnan

`ramakris@yahoo-inc.com`

Yahoo! Research

Databases and Data Mining

- What can database systems offer in the grand challenge of understanding and learning from the flood of data we've unleashed?
 - The plumbing
 - Scalability

Databases and Data Mining

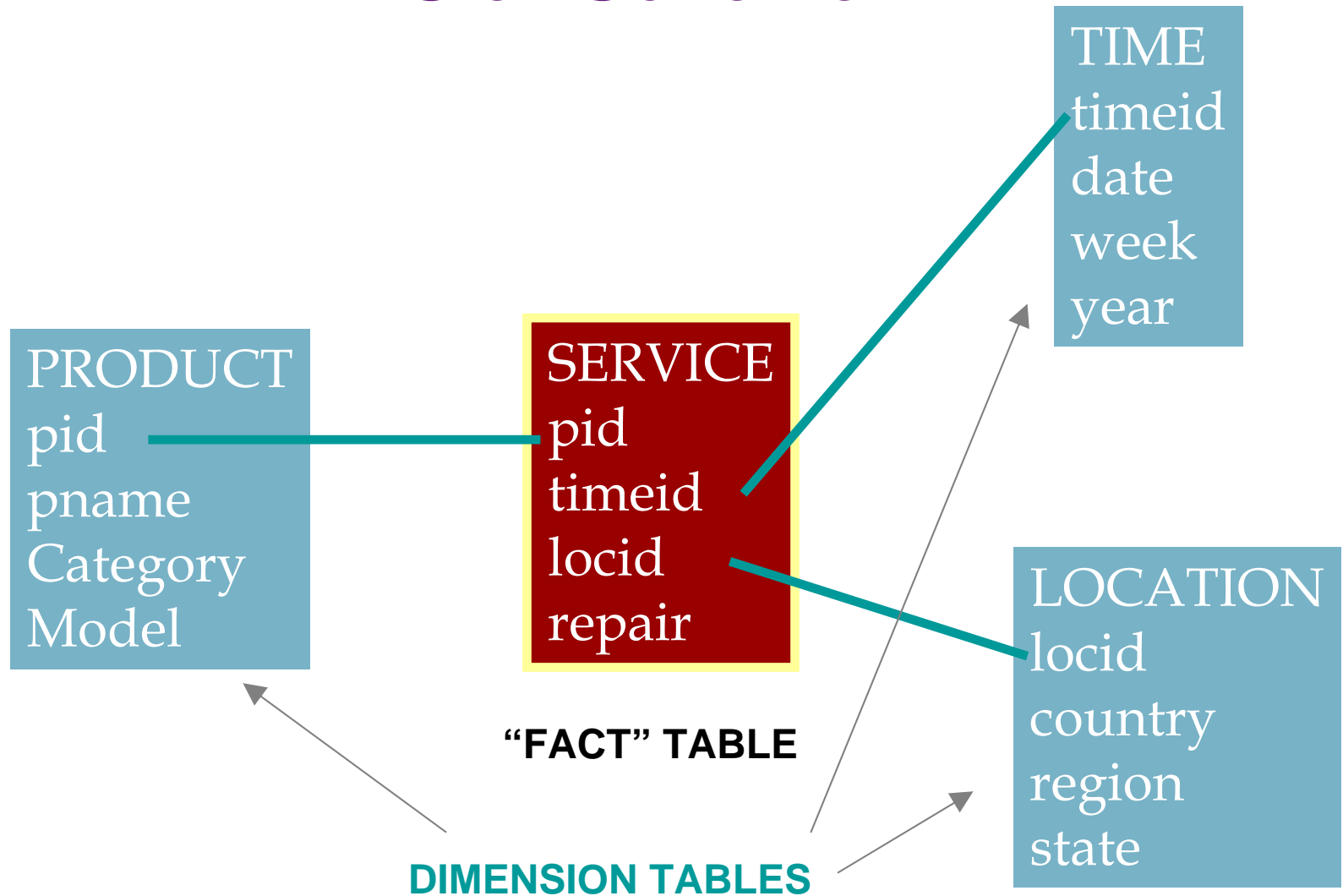
- What can database systems offer in the grand challenge of understanding and learning from the flood of data we've unleashed?
 - The plumbing
 - Scalability
 - Ideas!
 - Declarativeness
 - Compositionality
 - **Ways to conceptualize your data**

About this Talk

- Joint work with many people
- Common theme—multidimensional view of the data:
 - Helps handle imprecision
 - Analyzing imprecise and aggregated data
 - Defines candidate space of subsets for exploratory mining
 - Forecasting query results over “future data”
 - Using predictive models as summaries
 - Restricting candidate clusters
 - Potentially, space of “mining experiments”?

Background: The Multidimensional Data Model Cube Space

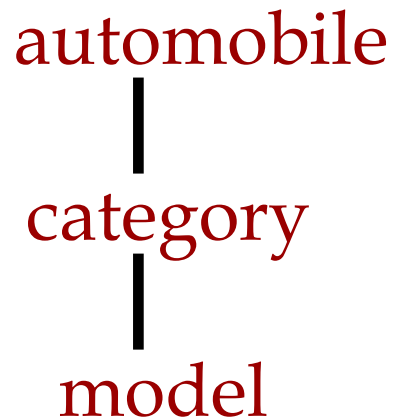
Star Schema



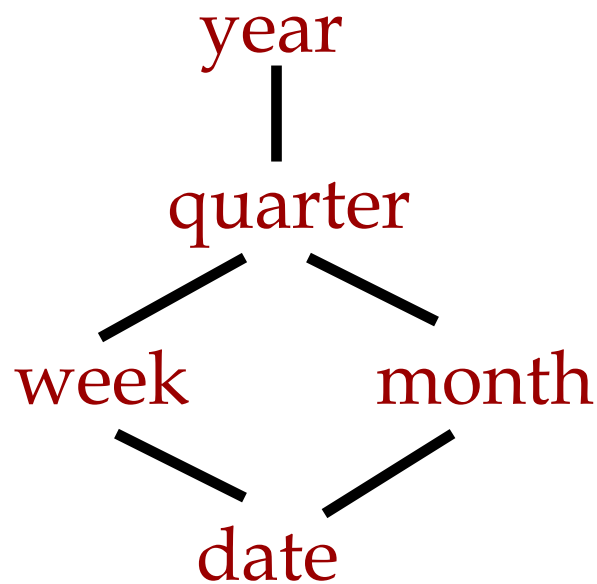
Dimension Hierarchies

- For each dimension, the set of values can be organized in a hierarchy:

PRODUCT



TIME



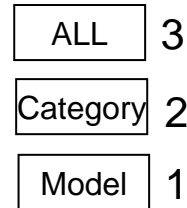
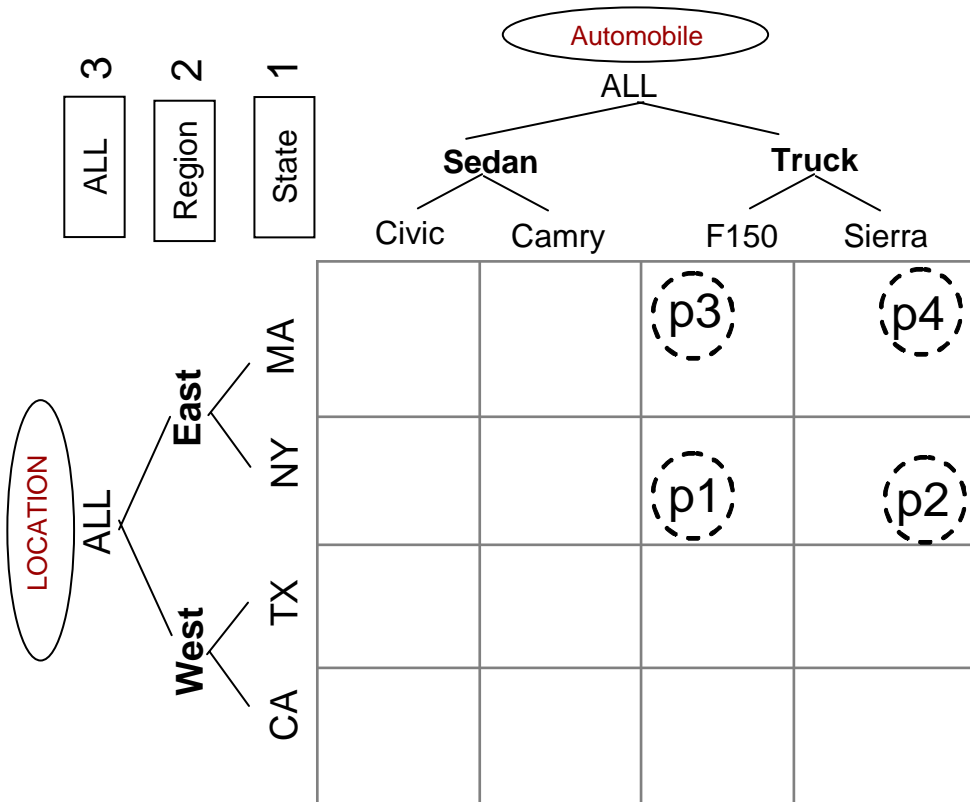
LOCATION



Multidimensional Data Model

- One fact table $\Delta=(\mathbf{X},\mathbf{M})$
 - $\mathbf{X}=X_1, X_2, \dots$ Dimension attributes
 - $\mathbf{M}=M_1, M_2, \dots$ Measure attributes
- Domain hierarchy for each dimension attribute:
 - Collection of domains $\text{Hier}(X_i) = (D_i^{(1)}, \dots, D_i^{(k)})$
 - The extended domain: $EX_i = \cup_{1 \leq k \leq t} DX_i^{(k)}$
- Value mapping function: $\gamma_{D_1 \rightarrow D_2}(x)$
 - e.g., $\gamma_{\text{month} \rightarrow \text{year}}(12/2005) = 2005$
 - Form the value hierarchy graph
 - Stored as dimension table attribute (e.g., week for a time value) or conversion functions (e.g., month, quarter)

Multidimensional Data



**DIMENSION
ATTRIBUTES**

<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200

Cube Space

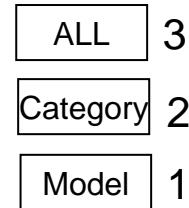
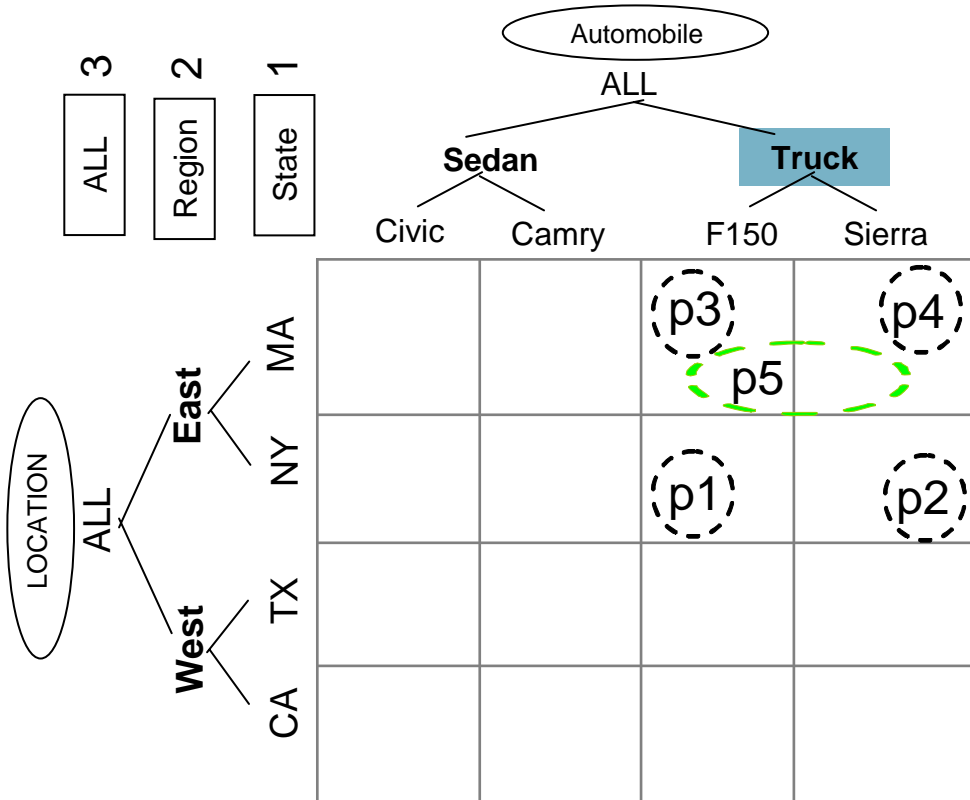
- Cube space: $C = EX_1 \times EX_2 \times \dots \times EX_d$
- Region: Hyper rectangle in cube space
 - $c = (v_1, v_2, \dots, v_d)$, $v_i \in EX_i$
- Region granularity:
 - $\text{gran}(c) = (d_1, d_2, \dots, d_d)$, $d_i = \text{Domain}(c.v_i)$
- Region coverage:
 - $\text{coverage}(c) = \text{all facts in } c$
- Region set: All regions with same granularity

OLAP Over Imprecise Data

with Doug Burdick, Prasad Deshpande, T.S. Jayram, and
Shiv Vaithyanathan

In VLDB 05, 06 joint work with IBM Almaden

Imprecise Data

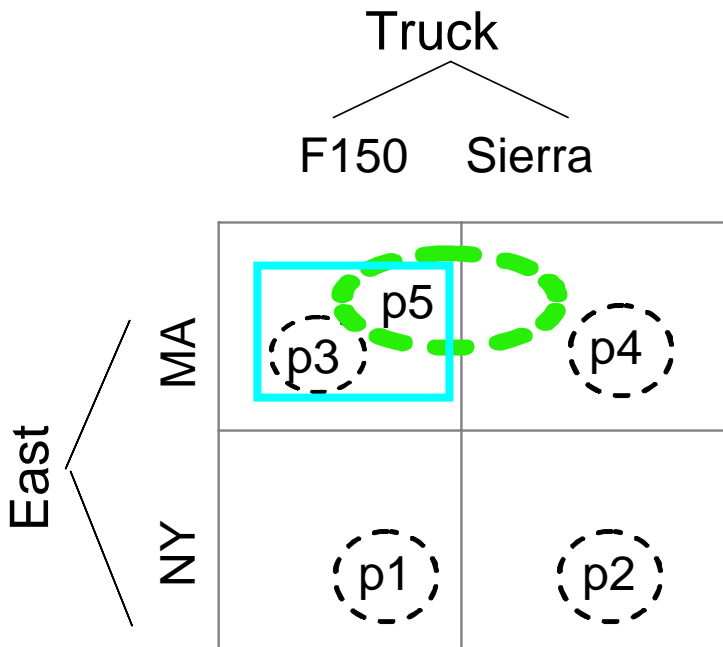


<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200
p5	Truck	MA	100

Querying Imprecise Facts

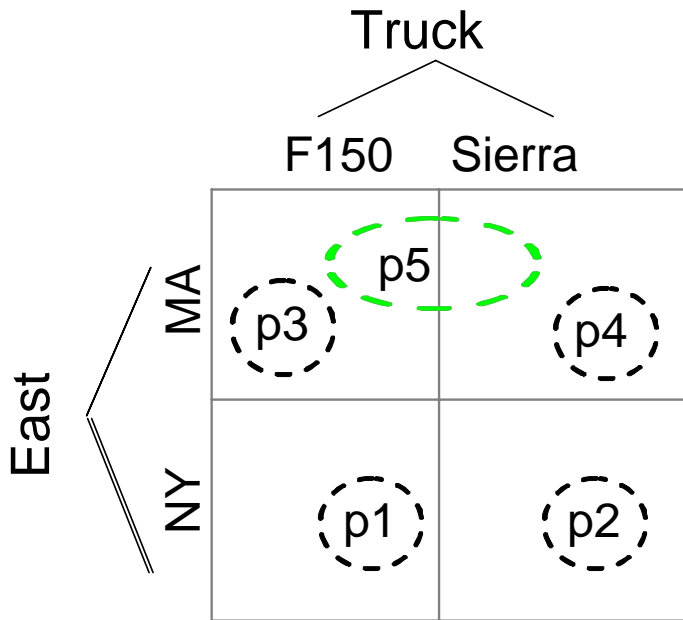
Auto = F150
 Loc = MA
 SUM(Repair) = ???

How do we treat p5?



FactID	Auto	Loc	Repair
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200
p5	Truck	MA	100

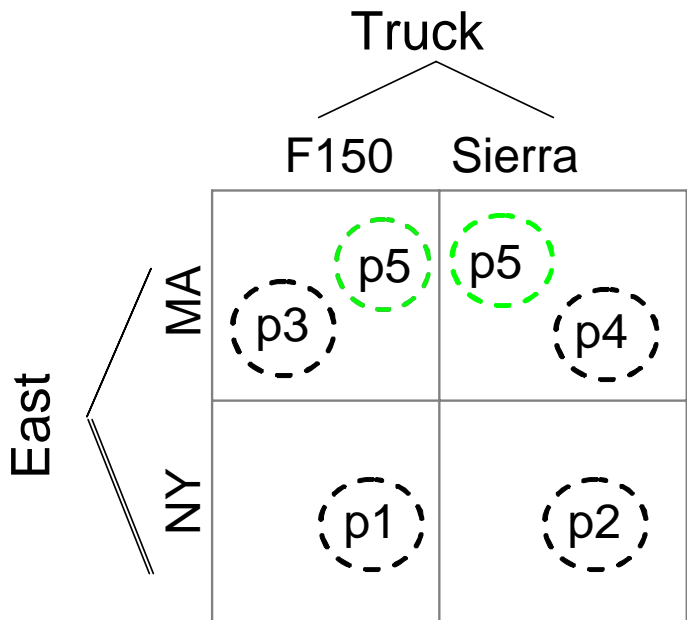
Allocation (1)



<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200
p5	Truck	MA	100

Allocation (2)

(Huh? Why 0.5 / 0.5?
- Hold on to that thought)

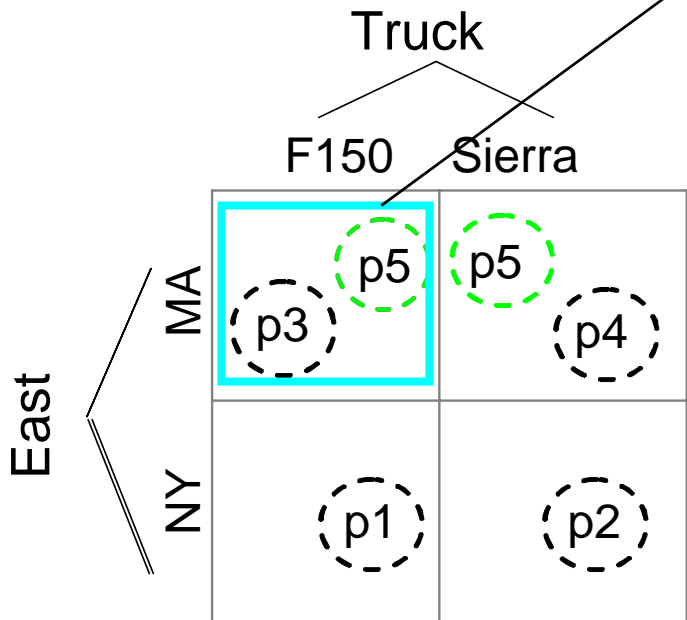


<i>ID</i>	<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>	<i>Weight</i>
1	p1	F150	NY	100	1.0
2	p2	Sierra	NY	500	1.0
3	p3	F150	MA	100	1.0
4	p4	Sierra	MA	200	1.0
5	p5	F150	MA	100	0.5
6	p5	Sierra	MA	100	0.5

Allocation (3)

Auto = F150
Loc = MA
SUM(Repair) = 150

Query the Extended Data Model!



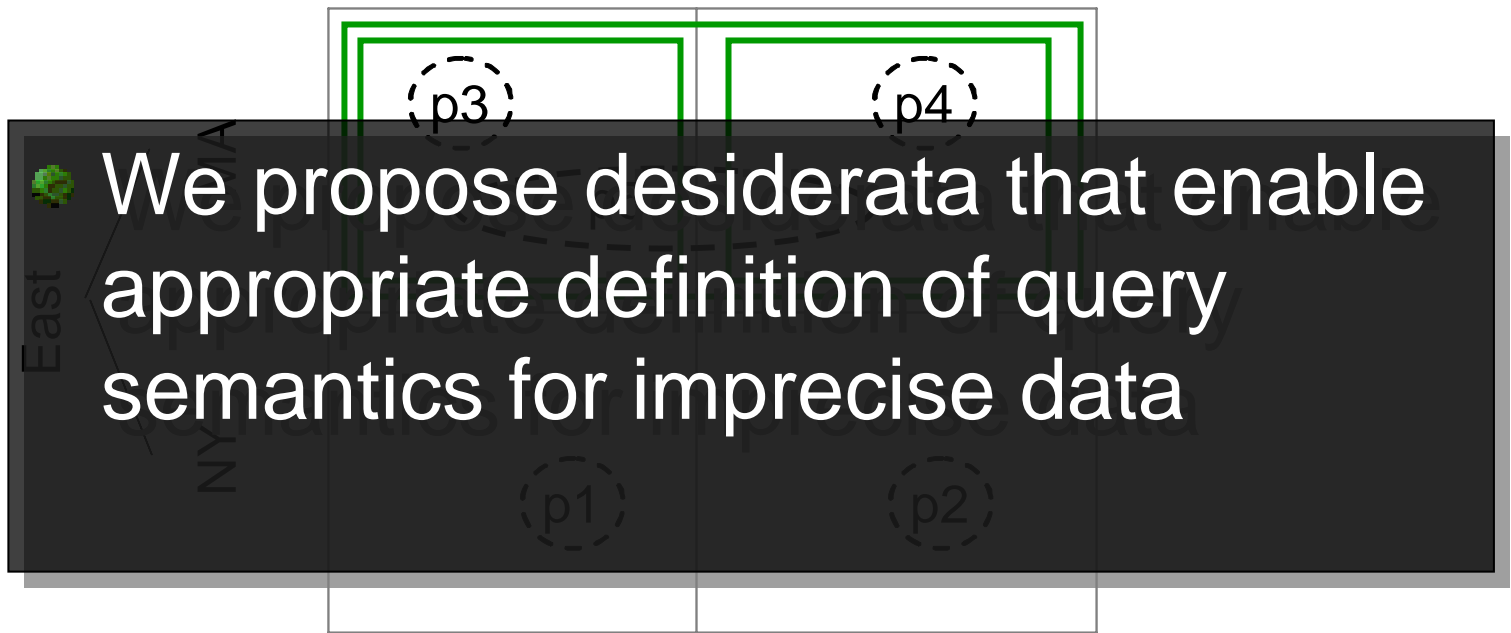
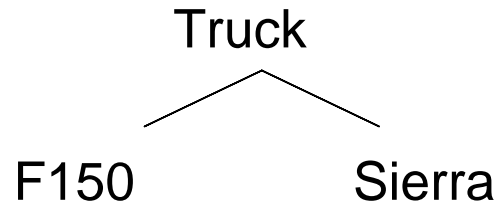
<i>ID</i>	<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>	<i>Weight</i>
1	p1	F150	NY	100	1.0
2	p2	Sierra	NY	500	1.0
3	p3	F150	MA	100	1.0
4	p4	Sierra	MA	200	1.0
5	p5	F150	MA	100	0.5
6	p5	Sierra	MA	100	0.5

Allocation Policies

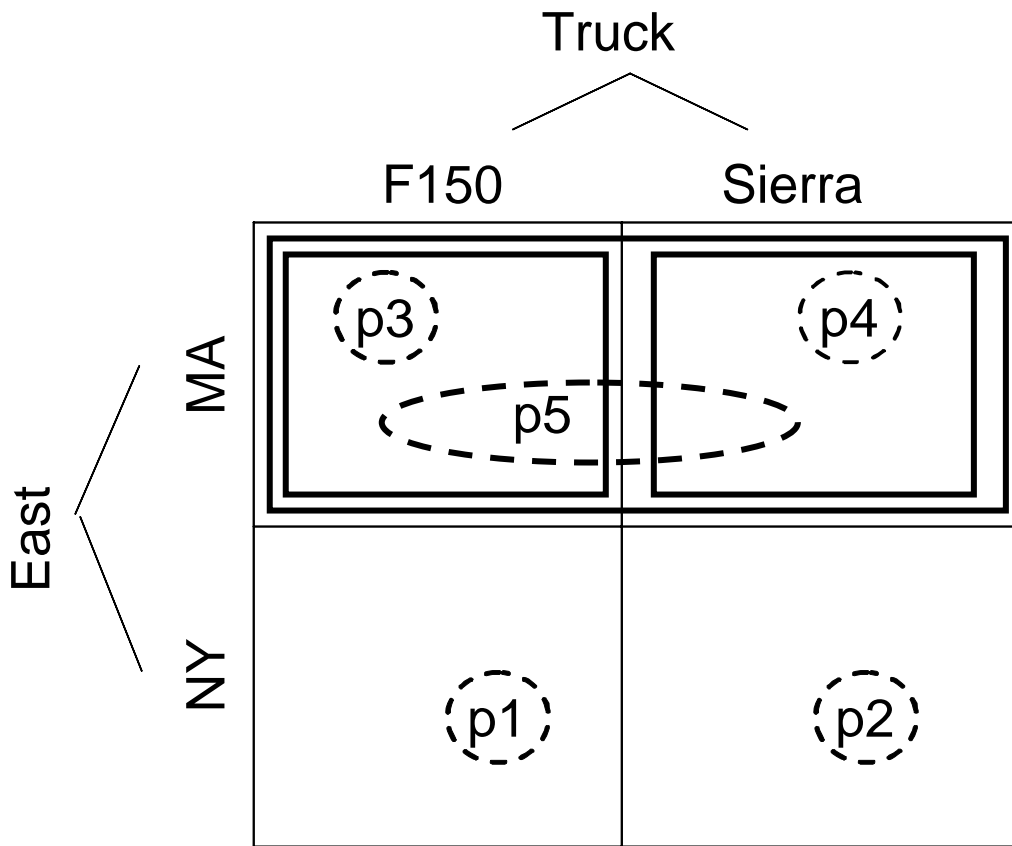
- Procedure for assigning allocation weights is referred to as an **allocation policy**
 - Each allocation policy uses different information to assign allocation weight
- **Key contributions:**
 - Appropriate characterization of the large space of allocation policies (VLDB 05)
 - Designing efficient algorithms for allocation policies that take into account the correlations in the data (VLDB 06)

Motivating Example

Query: COUNT



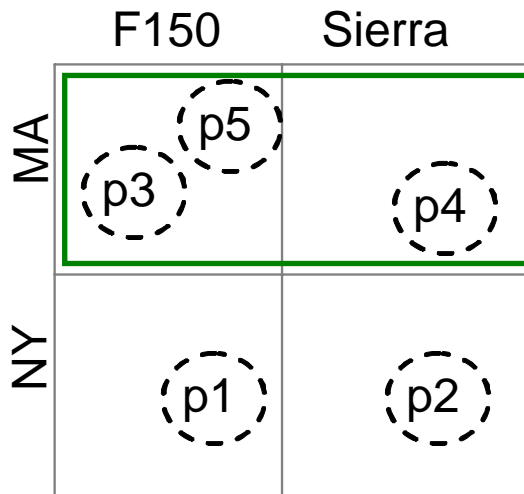
Desideratum I: Consistency



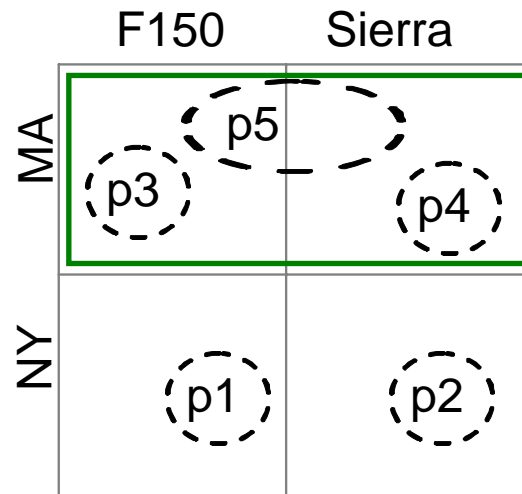
- Consistency specifies the relationship between answers to **related queries** on a **fixed data set**

Desideratum II: Faithfulness

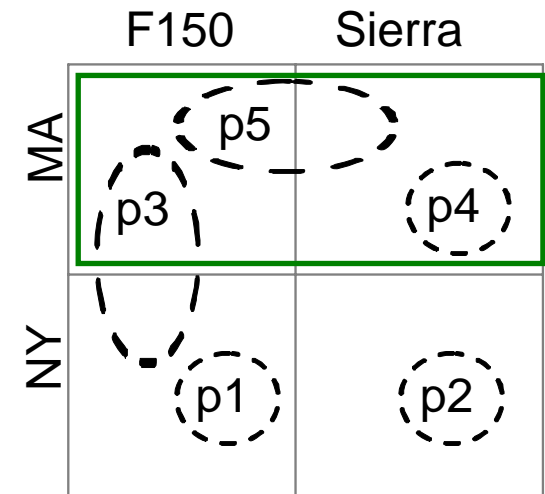
Data Set 1



Data Set 2

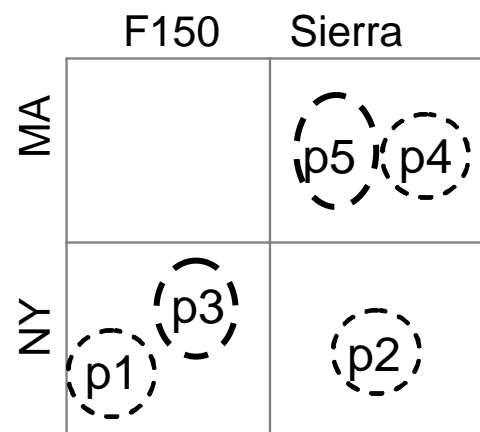
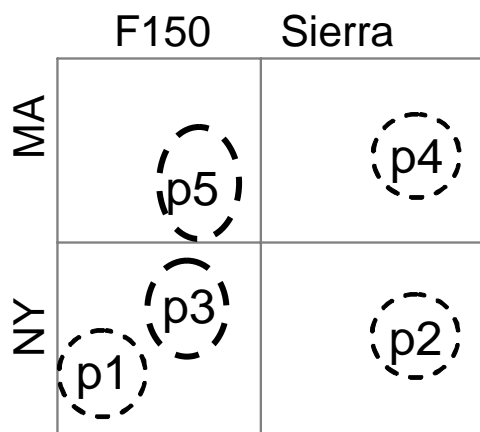
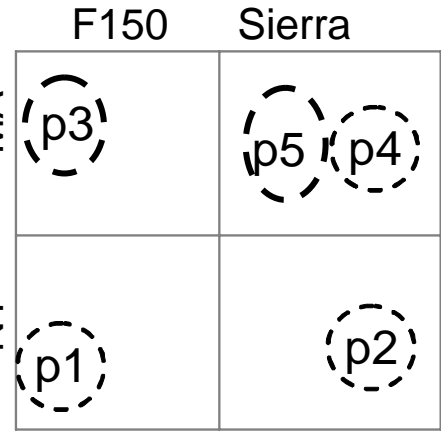
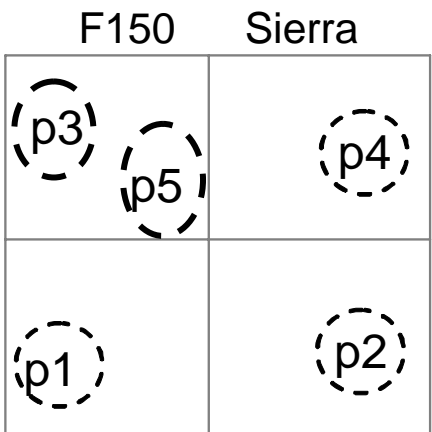
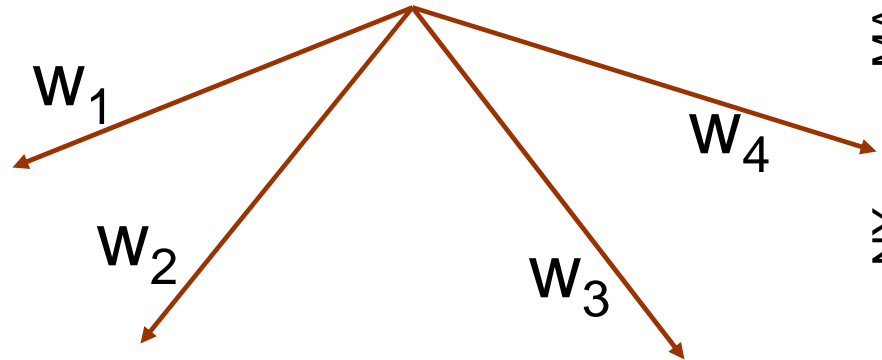
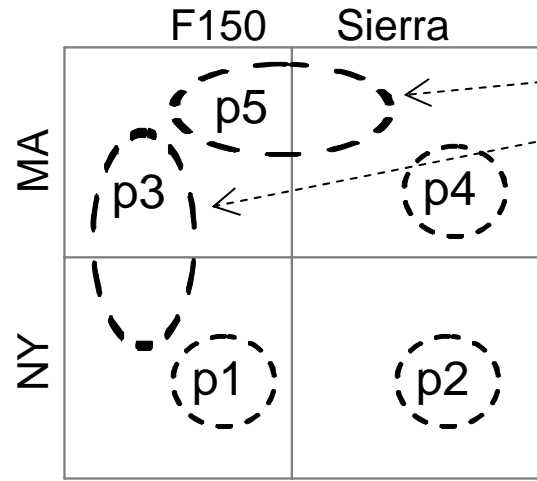


Data Set 3



- Faithfulness specifies the relationship between answers to a **fixed query** on **related data sets**

Imprecise facts lead to many possible worlds [Kripke63, ...]



Query Semantics

- Given all possible worlds together with their probabilities, queries are easily answered using expected values
 - But number of possible worlds is exponential!
- Allocation gives facts weighted assignments to possible completions, leading to an extended version of the data
 - Size increase is linear in number of (completions of) imprecise facts
 - Queries operate over this extended version

Bellwether Analysis: Global Aggregates from Local Regions

with Beechun Chen, Jude Shavlik, and Pradeep Tamma
In VLDB 06

Motivating Example

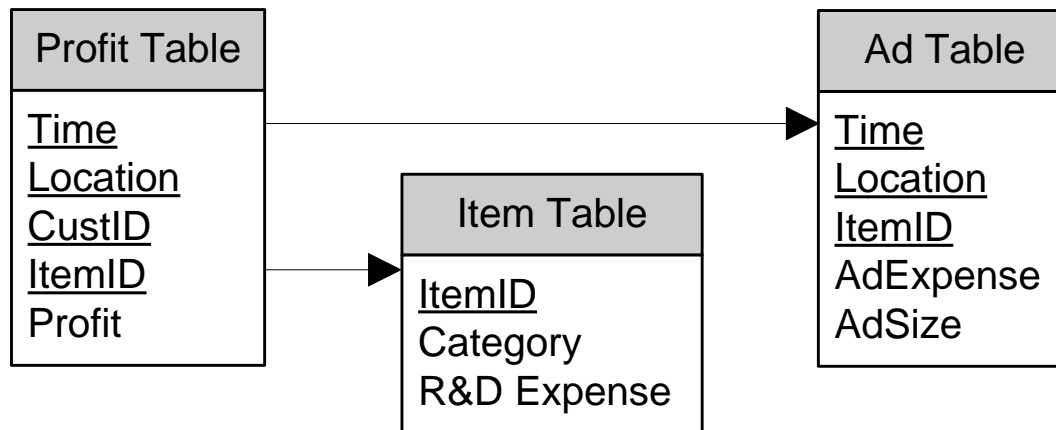
- A company wants to predict the first year worldwide profit of a new item (e.g., a new movie)
 - By looking at **features and profits of previous (similar) movies**, we predict **expected total profit** (1-year US sales) **for new movie**
 - **Wait a year and write a query! If you can't wait, stay awake ...**
 - The most predictive “features” may be based on sales data gathered by releasing the new movie in many “regions” (different locations over different time periods).
 - Example **“region-based” features**: 1st week sales in Peoria, week-to-week sales growth in Wisconsin, etc.
 - Gathering this data has a **cost** (e.g., marketing expenses, waiting time)
- **Problem statement**: Find the most predictive region features that can be obtained within a given “cost budget”

Key Ideas

- Large datasets are rarely labeled with the targets that we wish to learn to predict
 - But for the tasks we address, we can readily use OLAP queries to generate features (e.g., 1st week sales in Peoria) and even **targets** (e.g., profit) for mining
- We use data-mining models as building blocks in the mining process, rather than thinking of them as the end result
 - The central problem is to find data subsets (**“bellwether regions”**) that lead to predictive features which can be gathered at low cost for a new case

Motivating Example

- A company wants to predict the first year's worldwide profit for a new item, by using its historical database
- Database Schema:

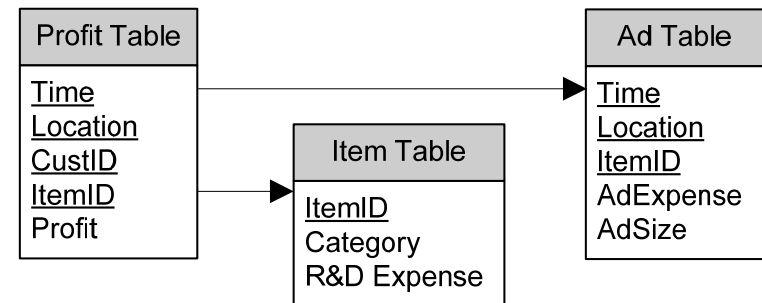


- The combination of the underlined attributes forms a key

A Straightforward Approach

- Build a regression model to predict item profit

By joining and aggregating tables in the **historical database** we can create a **training set**:



Item-table features			Target
ItemID	Category	R&D Expense	Profit
1	Laptop	500K	12,000K
2	Desktop	100K	8,000K
...

An Example regression model:
$$Profit = \beta_0 + \beta_1 Laptop + \beta_2 Desktop + \beta_3 RdExpense$$

- There is much room for accuracy improvement!

Using Regional Features

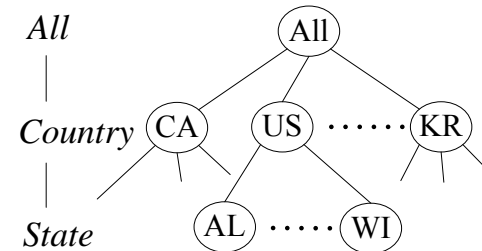
- Example region: [1st week, HK]
- **Regional features:**
 - **Regional Profit:** The 1st week profit in HK
 - **Regional Ad Expense:** The 1st week ad expense in HK
- A possibly more accurate model:

$$Profit_{[1\text{yr}, \text{All}]} = \beta_0 + \beta_1 Laptop + \beta_2 Desktop + \beta_3 RdExpense + \beta_4 \mathbf{Profit}_{[1\text{wk}, \text{HK}]} + \beta_5 \mathbf{AdExpense}_{[1\text{wk}, \text{HK}]}$$

- **Problem:** Which region should we use?
 - The smallest region that improves the accuracy the most
 - We give each candidate region a cost
 - The most “cost-effective” region is the **bellwether region**

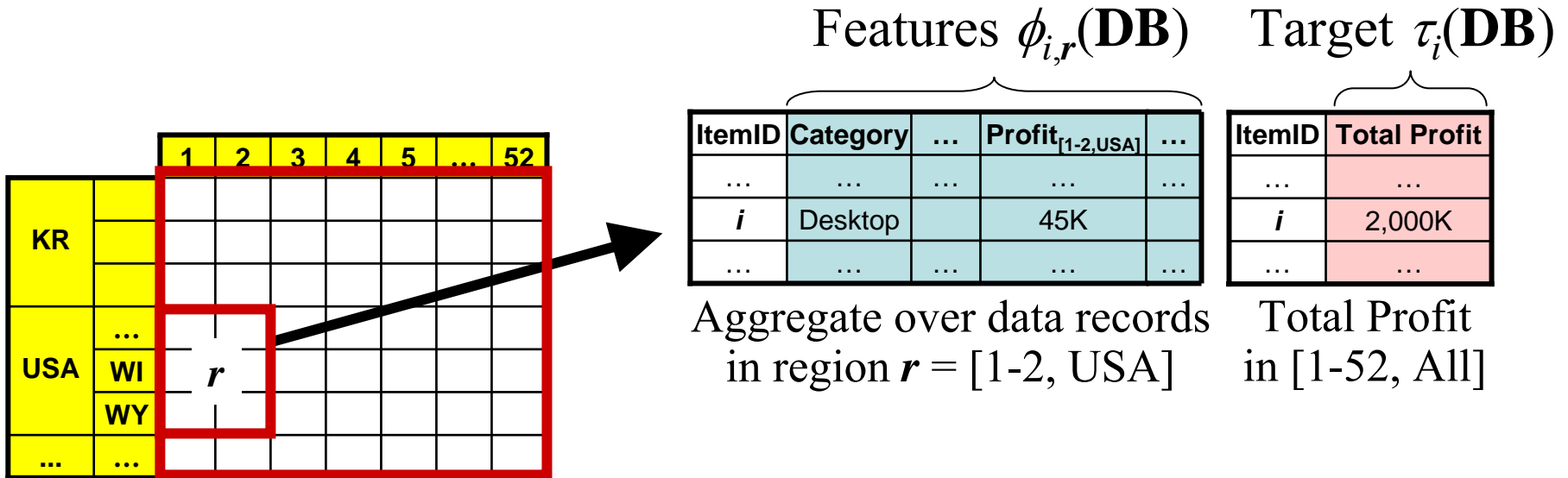
Basic Bellwether Problem

Location domain hierarchy



- Historical database: **DB**
- Training item set: **I**
- Candidate region set: **R**
 - E.g., $\{ [1-n \text{ week}, \text{Location}] \}$
- Target generation query: $\tau_i(\mathbf{DB})$ returns the target value of item $i \in \mathbf{I}$
 - E.g., $\alpha_{\text{sum(Profit)}} \sigma_{i, [1-52, \text{All}]} \text{ProfitTable}$
- Feature generation query: $\phi_{i,r}(\mathbf{DB})$, $i \in \mathbf{I}_r$ and $r \in \mathbf{R}$
 - \mathbf{I}_r : The set of items in region r
 - E.g., $[\text{Category}_i, \text{RdExpense}_i, \text{Profit}_{i, [1-n, \text{Loc}]}, \text{AdExpense}_{i, [1-n, \text{Loc}]}]$
- Cost query: $\kappa_r(\mathbf{DB})$, $r \in \mathbf{R}$, the cost of collecting data from r
- Predictive model: $h_r(\mathbf{x})$, $r \in \mathbf{R}$, trained on $\{(\phi_{i,r}(\mathbf{DB}), \tau_i(\mathbf{DB})) : i \in \mathbf{I}_r\}$
 - E.g., linear regression model

Basic Bellwether Problem

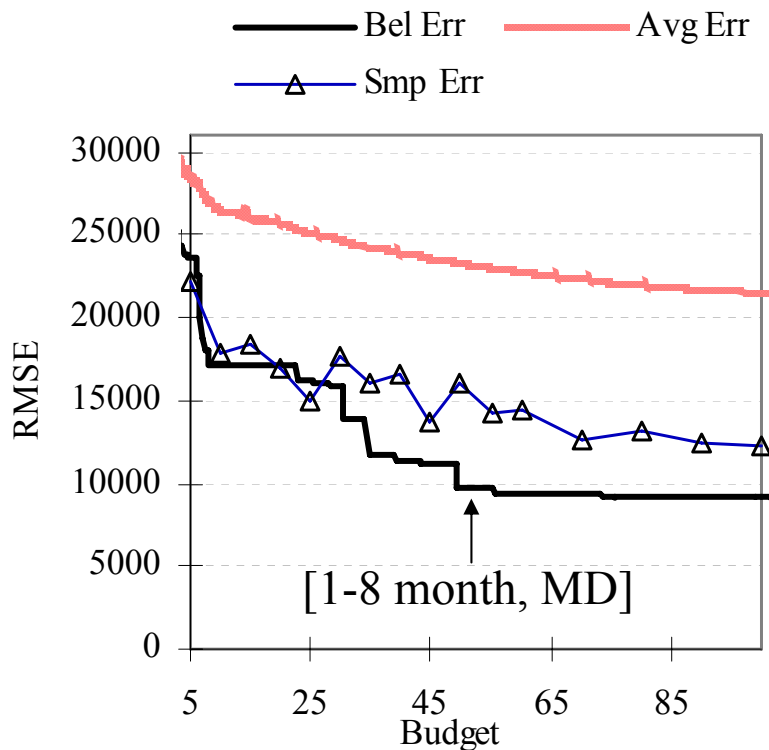


For each region r , build a predictive model $h_r(\mathbf{x})$; and then choose **bellwether region**:

- $Coverage(r) \equiv$ fraction of all items in region \geq minimum coverage support
- $Cost(r, \mathbf{DB}) \leq$ cost threshold
- $Error(h_r)$ is minimized

Experiment on a Mail Order Dataset

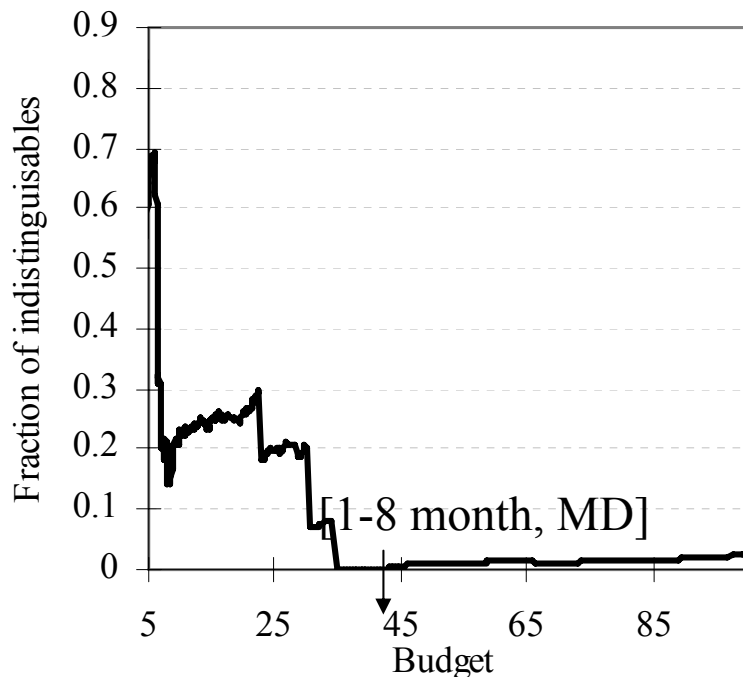
Error-vs-Budget Plot



- **Bel Err:** The error of the bellwether region found using a given budget
 - **Avg Err:** The average error of all the cube regions with costs under a given budget
 - **Smp Err:** The error of a set of randomly sampled (non-cube) regions with costs under a given budget
- (RMSE: Root Mean Square Error)

Experiment on a Mail Order Dataset

Uniqueness Plot



- **Y-axis:** Fraction of regions that are as good as the bellwether region
 - The fraction of regions that satisfy the constraints and have errors within the 99% confidence interval of the error of the bellwether region
- We have 99% confidence that that [1-8 month, MD] is a quite unusual bellwether region

Basic Bellwether Computation

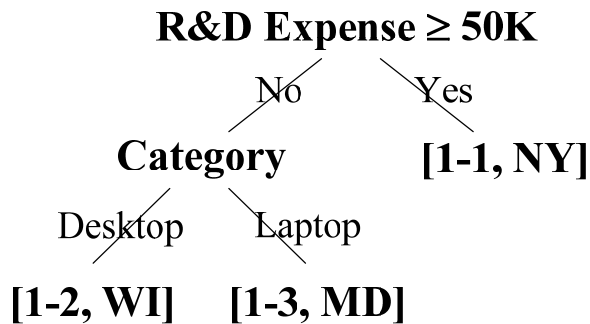
- OLAP-style bellwether analysis
 - **Candidate regions**: Regions in a data cube
 - Queries: OLAP-style aggregate queries
 - E.g., Sum(Profit) over a region
- Efficient computation:
 - Use **iceberg cube techniques to prune** infeasible regions (Beyer-Ramakrishnan, ICDE 99; Han-Pei-Dong-Wang SIGMOD 01)
 - Infeasible regions: Regions with cost $> B$ or coverage $< C$
 - **Share computation** by generating the features and target values for all the feasible regions all together
 - Exploit distributive and algebraic aggregate functions
 - Simultaneously generating all the features and target values reduces DB scans and repeated aggregate computation

		1	2	3	4	5	...	5
KR								
USA	...							
	WI							
	WY							
...	...							

Subset-Based Bellwether Prediction

- **Motivation:** Different subsets of items may have different bellwether regions
 - E.g., The bellwether region for laptops may be different from the bellwether region for clothes
- Two approaches:

Bellwether Tree



Bellwether Cube

		R&D Expenses			
		Low	Medium	High	
Category	Software	OS	[1-3,CA]	[1-1,NY]	[1-2,CA]
	
	Hardware	Laptop	[1-4,MD]	[1-1, NY]	[1-3,WI]
	
	

Characteristics of Bellwether Trees & Cubes

Dataset generation:

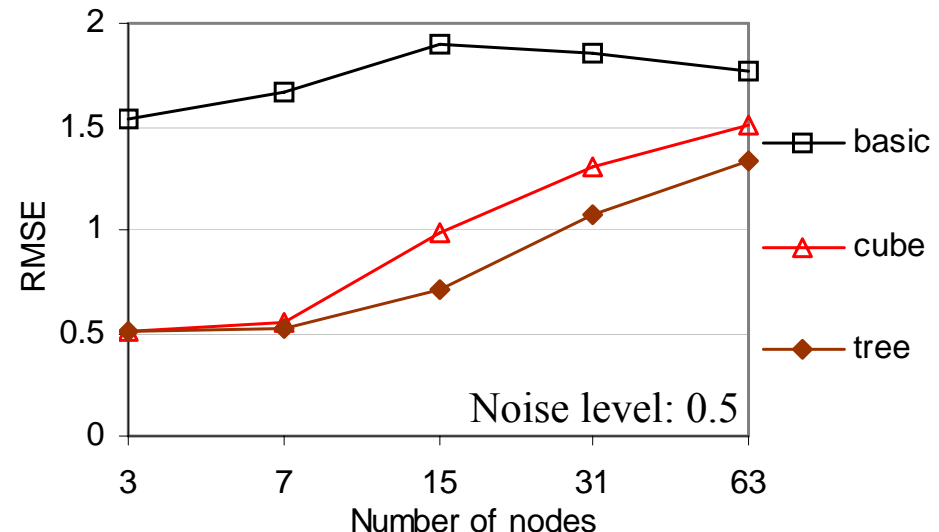
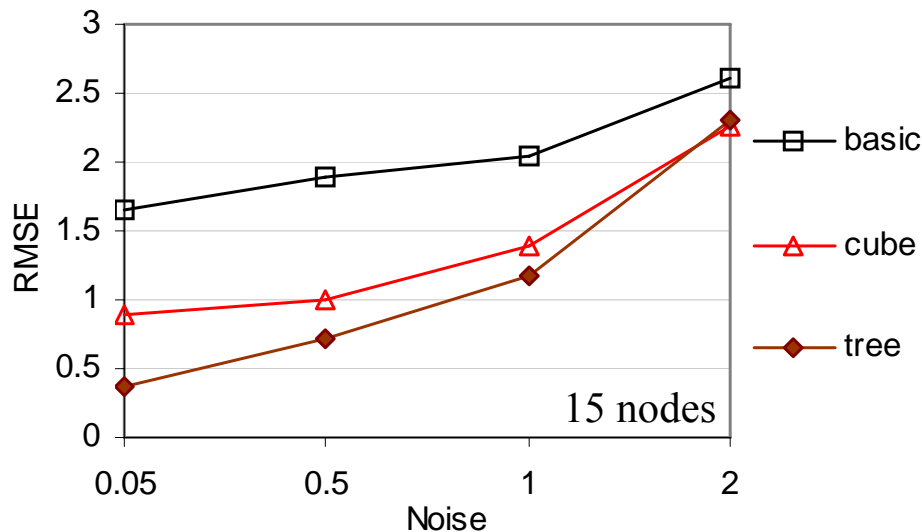
- Use random tree to generate different bellwether regions for different subset of items

Parameters:

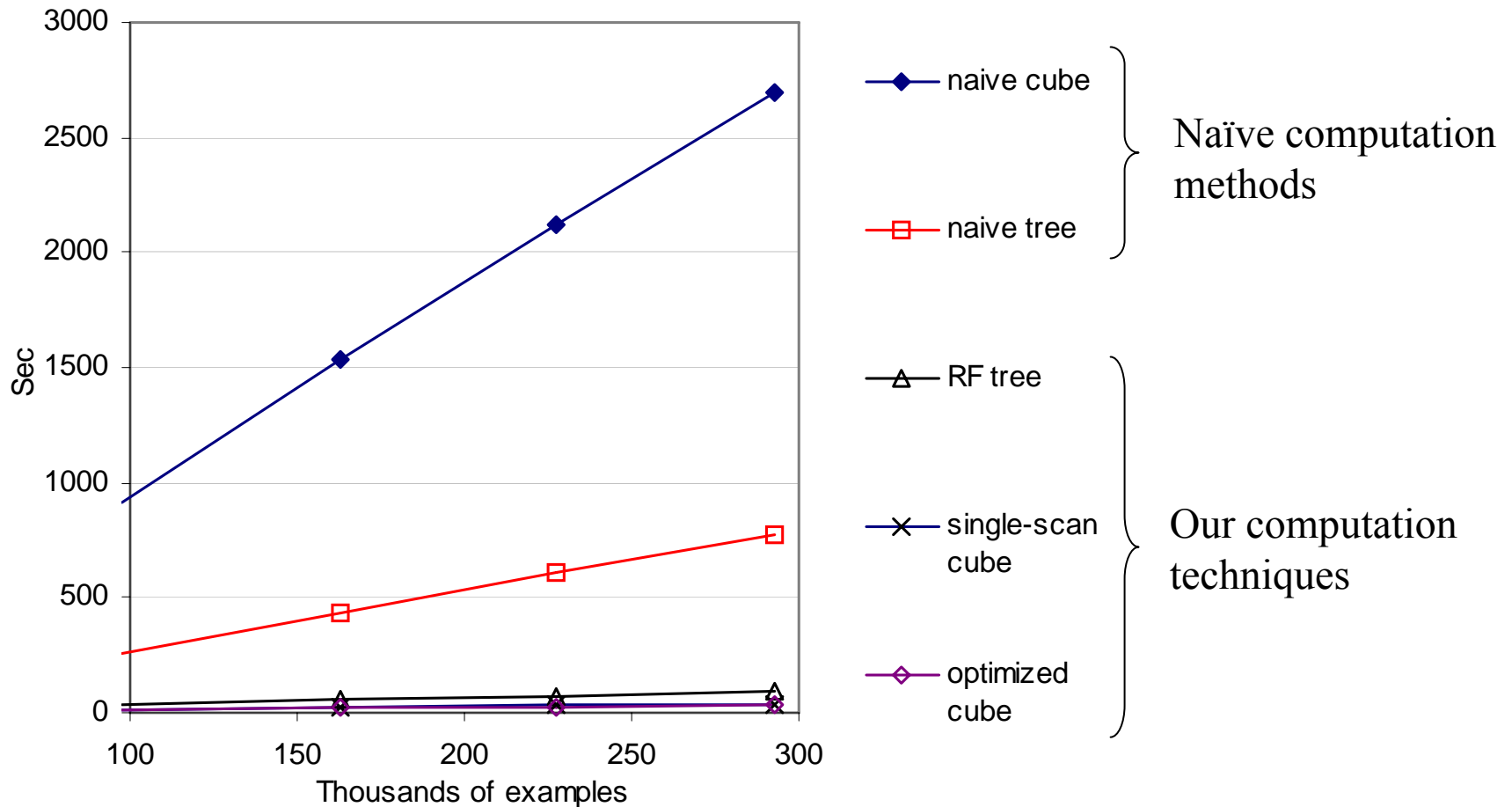
- Noise
- Concept complexity: # of tree nodes

Result:

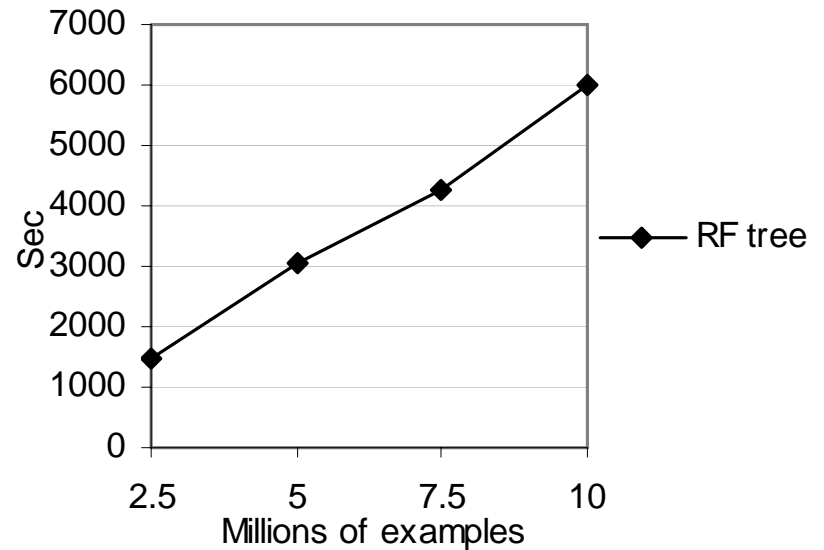
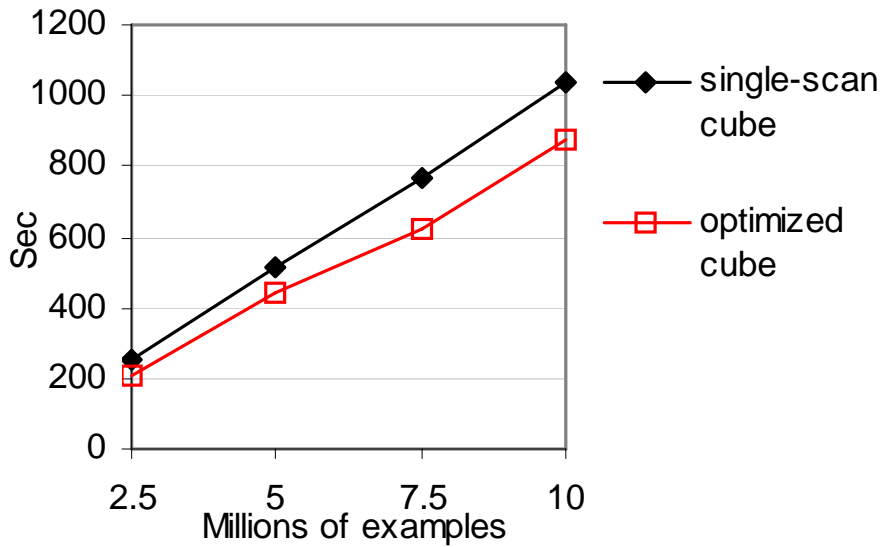
- Bellwether trees & cubes have better accuracy than basic bellwether search
- Increase noise \Rightarrow increase error
- Increase complexity \Rightarrow increase error



Efficiency Comparison



Scalability



Exploratory Mining: Prediction Cubes

with Beechun Chen, Lei Chen, and Yi Lin
In VLDB 05; EDAM Project

The Idea

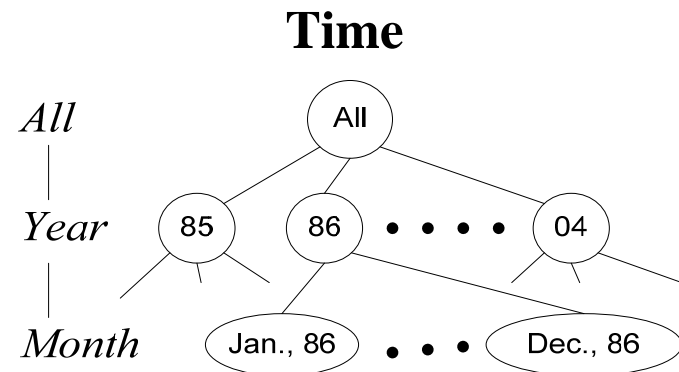
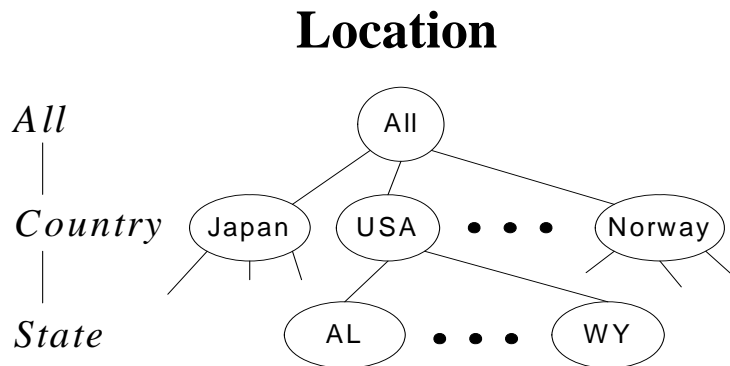
- Build OLAP data cubes in which cell values represent **decision/prediction behavior**
 - In effect, build a tree for each cell/region in the cube— observe that this is **not** the same as a collection of trees used in an ensemble method!
 - The idea is simple, but it leads to promising data mining tools
 - **Ultimate objective:** Exploratory analysis of the entire space of “data mining choices”
 - Choice of algorithms, data conditioning parameters ...

Example (1/7): Regular OLAP

Goal: Look for patterns of unusually high numbers of applications:

Z: Dimensions Y: Measure

Location	Time	# of App.
...
AL, USA	Dec, 04	2
...
WY, USA	Dec, 04	3



Example (2/7): Regular OLAP

Goal: Look for patterns of unusually high numbers of applications:

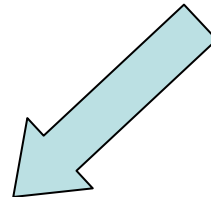
Z: Dimensions Y: Measure

Location	Time	# of App.
...
AL, USA	Dec, 04	2
...
WY, USA	Dec, 04	3

Coarser regions

	04	03	...
CA	100	90	...
USA	80	90	...
...

 Roll up



	2004			2003			...
	Jan	...	Dec	Jan	...	Dec	...
CA	30	20	50	25	30
USA	70	2	8	10
...

Drill down


		2004			...
		Jan	...	Dec	...
CA	AB	20	15	15	...
	...	5	2	20	...
	YT	5	3	15	...
USA	AL	55
	...	5
	WY	10
...

Cell value: Number of loan applications

Finer regions

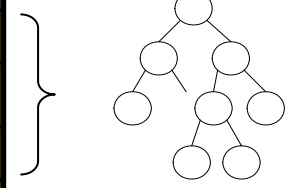
Example (3/7): Decision Analysis

Goal: Analyze a bank's loan **decision process** w.r.t. two dimensions: *Location* and *Time*

Fact table **D**

Z: Dimensions **X:** Predictors **Y:** Class

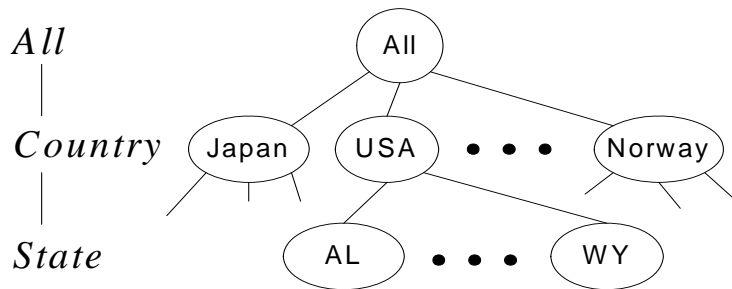
Location	Time	Race	Sex	...	Approval
AL, USA	Dec, 04	White	M	...	Yes
...
WY, USA	Dec, 04	Black	F	...	No



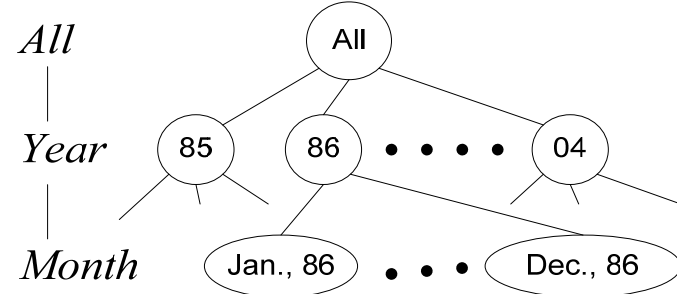
Cube subset

Model $h(\mathbf{X}, \sigma_{\mathbf{Z}}(\mathbf{D}))$
E.g., decision tree

Location



Time



Example (3/7): Decision Analysis

- Are there branches (and time windows) where approvals were closely tied to sensitive attributes (e.g., race)?
 - Suppose you partitioned the training data by location and time, chose the partition for a given branch and time window, and built a classifier. You could then ask, “Are the predictions of this classifier closely correlated with race?”
- Are there branches and times with decision making reminiscent of 1950s Alabama?
 - Requires comparison of classifiers trained using different subsets of data.

Example (4/7): Prediction Cubes

	2004			2003			...
	Jan	...	Dec	Jan	...	Dec	...
CA	0.4	0.8	0.9	0.6	0.8
USA	0.2	0.3	0.5
...

Data $\sigma_{[USA, Dec\ 04]}(\mathbf{D})$

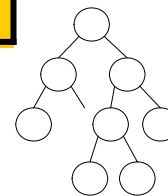
Location	Time	Race	Sex	...	Approval
AL, USA	Dec, 04	White	M	...	Y
...
WY, USA	Dec, 04	Black	F	...	N

1. Build a model using data from USA in Dec., 1985
2. Evaluate that model

Measure in a cell:

- **Accuracy** of the model
- **Predictiveness** of *Race* measured based on that model
- **Similarity** between that model and a given model

Model $h(\mathbf{X}, \sigma_{[USA, Dec\ 04]}(\mathbf{D}))$
E.g., decision tree



Example (5/7): Model-Similarity

Given:

- Data table **D**
- Target model $h_0(\mathbf{X})$
- Test set Δ w/o labels

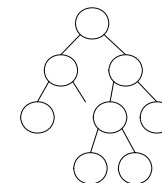
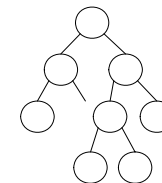
Data table **D**

Location	Time	Race	Sex	...	Approval
AL, USA	Dec, 04	White	M	...	Yes
...
WY, USA	Dec, 04	Black	F	...	No

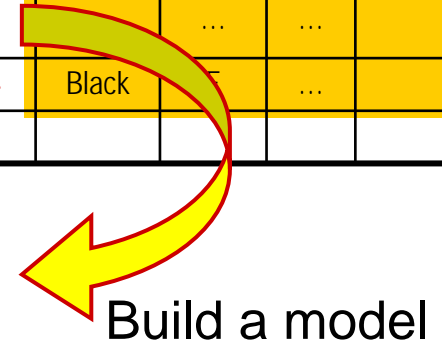
	2004			2003			...
	Jan	...	Dec	Jan	...	Dec	...
CA	0.4	0.2	0.3	0.6	0.5
USA	0.2	0.3	0.9
...

Level: [Country, Month]

Similarity



$h_0(\mathbf{X})$



Build a model

Race	Sex		
White	F	Yes	Yes
...
Black	M	No	Yes

Test set Δ

The loan decision process in **USA during Dec 04** was **similar to** a discriminatory decision model

Example (6/7): Predictiveness

Given:

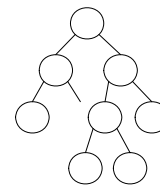
- Data table **D**
- Attributes **V**
- Test set Δ w/o labels

	2004			2003			...
	Jan	...	Dec	Jan	...	Dec	...
CA	0.4	0.2	0.3	0.6	0.5
USA	0.2	0.3	0.9
...

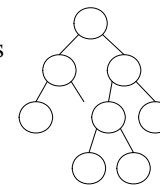
Level: [Country, Month]

Data table D

Location	Time	Race	Sex	...	Approval
AL, USA	Dec, 04	White	M	...	Yes
...
WY, USA	Dec, 04	Black	F	...	No



Yes
No
.
Yes

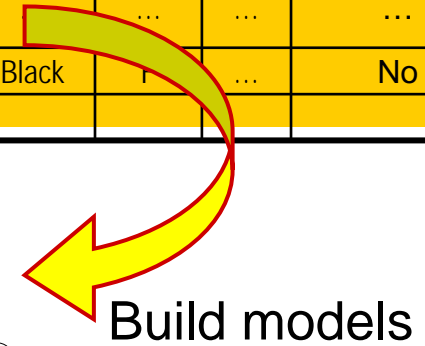


Yes
No
.
No

$h(X)$

$h(X-V)$

Predictiveness of **V**



Build models

Race	Sex	...
White	F	...
...
Black	M	...

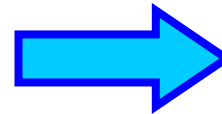
Test set Δ

Race was an important predictor of loan approval decision in **USA during Dec 04**

Example (7/7): Prediction Cube

	2004			2003			...
	Jan	...	Dec	Jan	...	Dec	...
CA	0.4	0.1	0.3	0.6	0.8
USA	0.7	0.4	0.3	0.3
...

Roll up



	04	03	...
CA	0.3	0.2	...
USA	0.2	0.3	...
...

Cell value: Predictiveness of *Race*



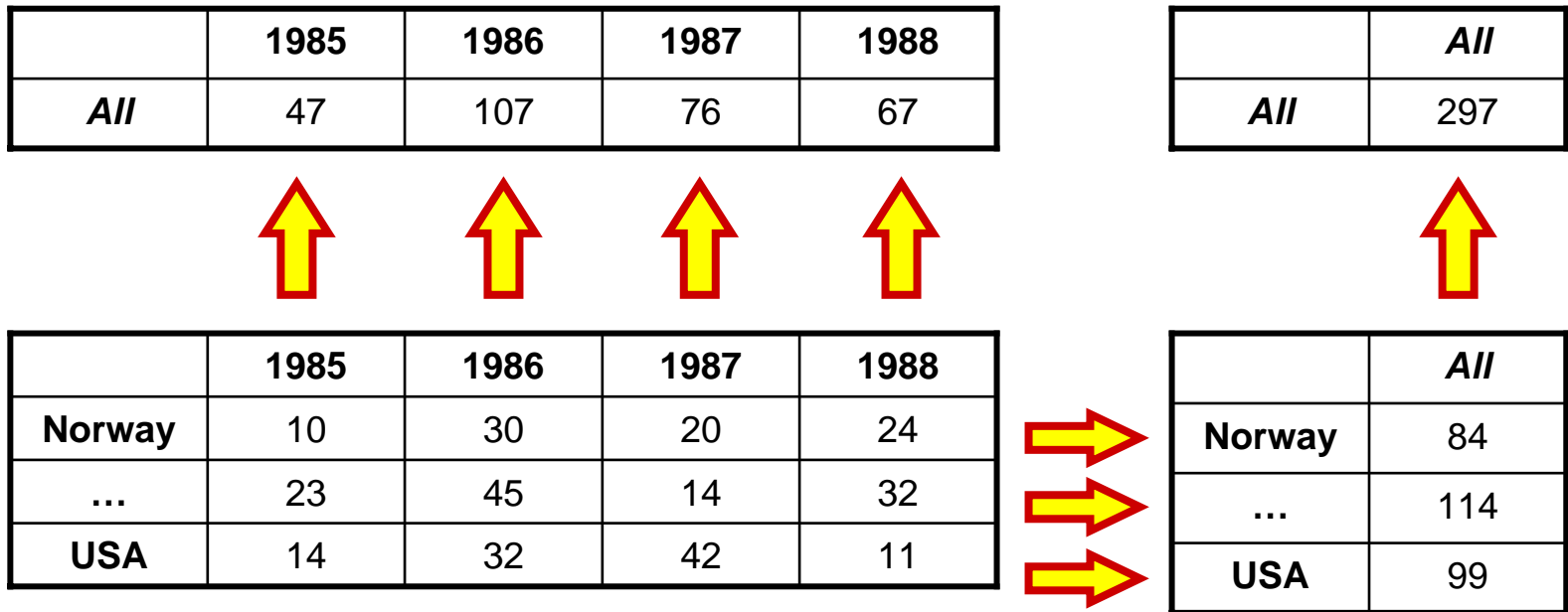
Drill down

		2004			2003			...
		Jan	...	Dec	Jan	...	Dec	...
CA	AB	0.4	0.2	0.1	0.1	0.2
	...	0.1	0.1	0.3	0.3
	YT	0.3	0.2	0.1	0.2
USA	AL	0.2	0.1	0.2
	...	0.3	0.1	0.1
	WY	0.9	0.7	0.8
...

Efficient Computation

- Reduce prediction cube computation to data cube computation
 - Represent a data-mining model as a distributive or algebraic (bottom-up computable) aggregate function, so that data-cube techniques can be directly applied

Bottom-Up Data Cube Computation



Cell Values: Numbers of loan applications

Functions on Sets

- Bottom-up computable functions: Functions that can be computed using only summary information
- **Distributive** function: $\alpha(X) = F(\{\alpha(X_1), \dots, \alpha(X_n)\})$
 - $X = X_1 \cup \dots \cup X_n$ and $X_i \cap X_j = \emptyset$
 - E.g., $Count(X) = Sum(\{Count(X_1), \dots, Count(X_n)\})$
- **Algebraic** function: $\alpha(X) = F(\{G(X_1), \dots, G(X_n)\})$
 - $G(X_i)$ returns a length-fixed vector of values
 - E.g., $Avg(X) = F(\{G(X_1), \dots, G(X_n)\})$
 - $G(X_i) = [Sum(X_i), Count(X_i)]$
 - $F(\{[s_1, c_1], \dots, [s_n, c_n]\}) = Sum(\{s_j\}) / Sum(\{c_j\})$

Scoring Function

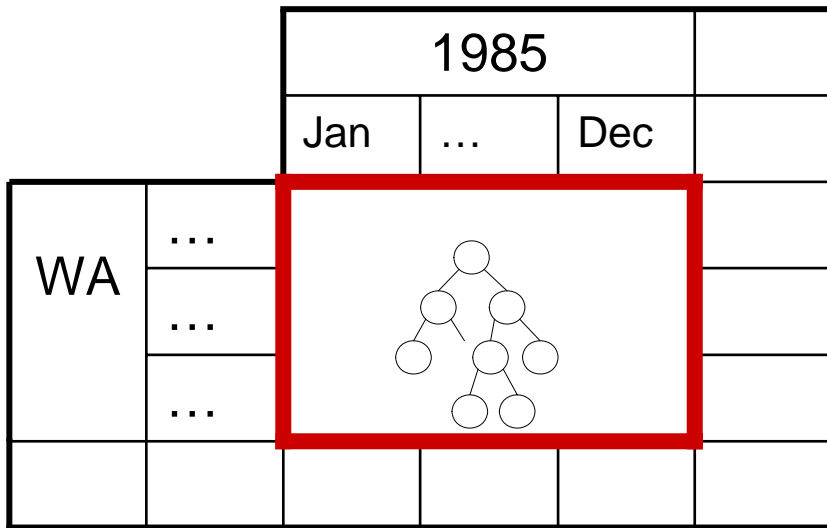
- Represent a model as a function of sets
- Conceptually, a machine-learning model $h(\mathbf{X}; \sigma_{\mathbf{Z}}(\mathbf{D}))$ is a scoring function $Score(y, \mathbf{x}; \sigma_{\mathbf{Z}}(\mathbf{D}))$ that gives each class y a score on test example \mathbf{x}
 - $h(\mathbf{x}; \sigma_{\mathbf{Z}}(\mathbf{D})) = \operatorname{argmax}_y Score(y, \mathbf{x}; \sigma_{\mathbf{Z}}(\mathbf{D}))$
 - $Score(y, \mathbf{x}; \sigma_{\mathbf{Z}}(\mathbf{D})) \approx p(y | \mathbf{x}, \sigma_{\mathbf{Z}}(\mathbf{D}))$
 - $\sigma_{\mathbf{Z}}(\mathbf{D})$: The set of training examples (a cube subset of \mathbf{D})

Machine-Learning Models

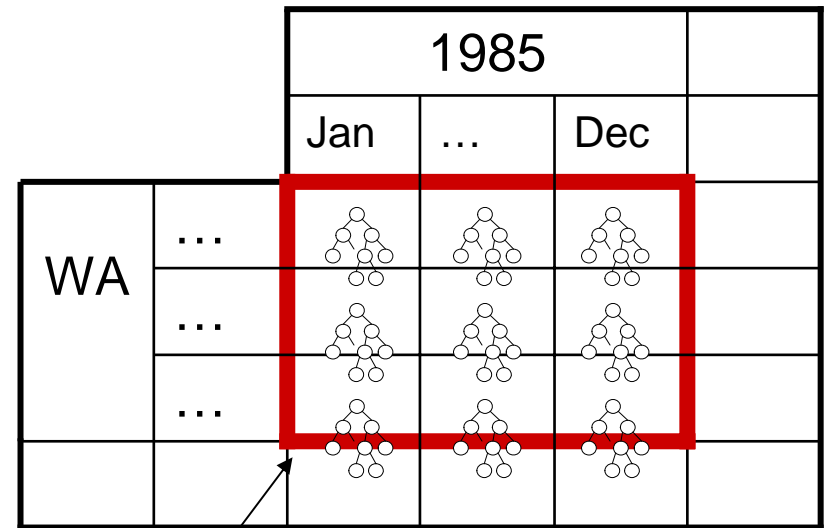
- Naïve Bayes:
 - Scoring function: algebraic
- Kernel-density-based classifier:
 - Scoring function: distributive
- Decision tree, random forest:
 - Neither distributive, nor algebraic
- PBE: Probability-based ensemble (new)
 - To make any machine-learning model distributive
 - Approximation

Probability-Based Ensemble

Decision tree on [WA, 85]

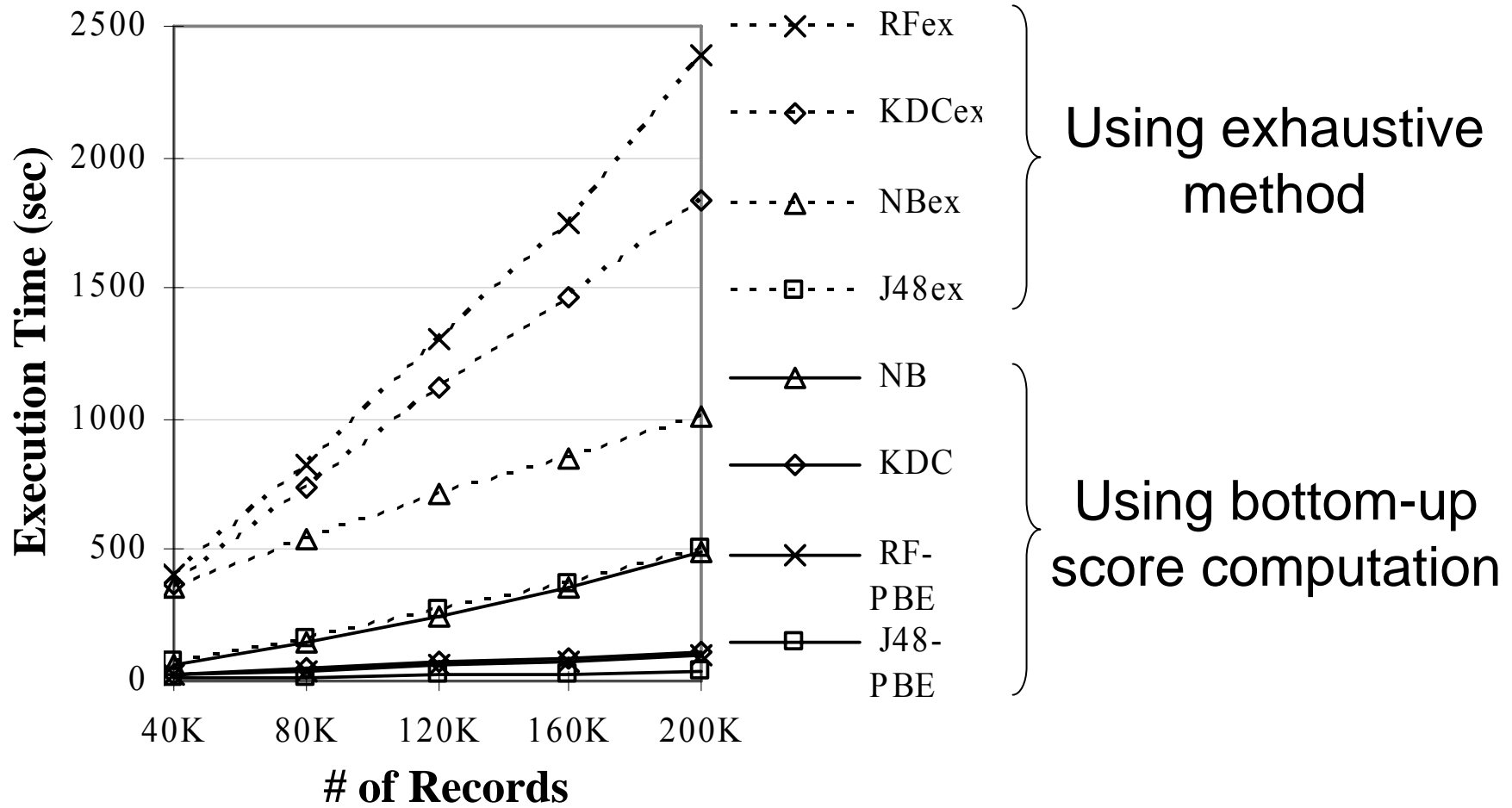


PBE version of decision tree on [WA, 85]



Decision trees built on the lowest-level cells

Efficiency Comparison



Conclusions

Related Work: Building models on OLAP Results

- Multi-dimensional regression [Chen, VLDB 02]
 - Goal: Detect changes of trends
 - Build linear regression models for cube cells
- Step-by-step regression in stream cubes [Liu, PAKDD 03]
- Loglinear-based quasi cubes [Barbara, J. IIS 01]
 - Use loglinear model to approximately compress dense regions of a data cube
- NetCube [Margaritis, VLDB 01]
 - Build Bayes Net on the entire dataset of approximate answer count queries

Related Work (Contd.)

- Cubegrades [Imielinski, J. DMKD 02]
 - Extend cubes with ideas from association rules
 - How does the measure change when we rollup or drill down?
- Constrained gradients [Dong, VLDB 01]
 - Find pairs of similar cell characteristics associated with big changes in measure
- User-cognizant multidimensional analysis [Sarawagi, VLDBJ 01]
 - Help users find the most informative unvisited regions in a data cube using max entropy principle
- Multi-Structural DBs [Fagin et al., PODS 05, VLDB 05]

Take-Home Messages

- Promising exploratory data analysis paradigm:
 - Can use **models** to identify interesting subsets
 - Concentrate only on subsets in **cube space**
 - Those are meaningful subsets, tractable
 - **Precompute** results and provide the users with an **interactive** tool
- A simple way to plug “something” into cube-style analysis:
 - Try to describe/approximate “something” by a distributive or algebraic function

Big Picture

- **Why stop with decision behavior?** Can apply to other kinds of analyses too
- **Why stop at browsing?** Can mine prediction cubes in their own right
- **Exploratory analysis of mining space:**
 - Dimension attributes can be parameters related to algorithm, data conditioning, etc.
 - Tractable evaluation is a challenge:
 - Large number of “dimensions”, real-valued dimension attributes, difficulties in compositional evaluation
 - Active learning for experiment design, extending compositional methods