On-Line Analytical Processing (OLAP)

CSE 6331 / CSE 6362

Data Mining

Fall 1999

Diane J. Cook

Traditional OLTP

- DBMS used for on-line transaction processing (OLTP)
  - order entry: pull up order xx-yy-zz and update status field
  - banking: transfer $100 from account no XXX to account no. YYY
    - clerical data processing tasks
    - detailed up-to-date data
    - structured, repetitive tasks
    - short transactions are the unit of work
    - read and/or update a few records
    - isolation, recovery and integrity are critical

OLTP vs. OLAP

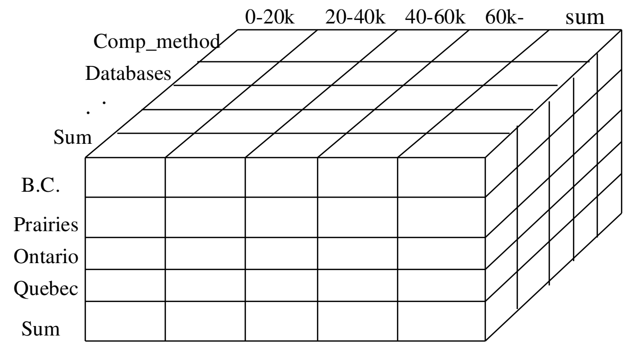
	OLTP	OLAP
users	Clerk, IT professional	Knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
access	read/write index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

On-Line Analytic Processing (OLAP)

- Analyze (summarize, consolidate, view, process) along multiple levels of abstraction and from different angles
  - Data in databases are often expressed at primitive levels.
  - Knowledge is usually expressed at high levels.
  - Data may imply concepts at multiple levels:

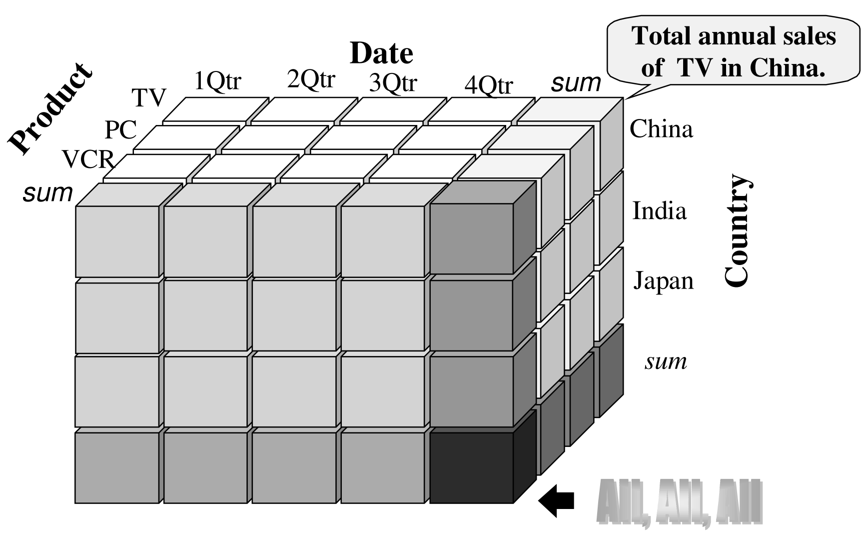
    Tom Jackson $\in$ CS grad $\subset$ student $\subset$ person.
- Mining knowledge just at single abstraction level?
  - Too low level? -- Raw data or weak rules.
  - Too high level? -- Not novel, common sense?
- Mining knowledge at multiple levels:
  - Provides different views and different abstractions.
  - Progressively focuses on "interesting" spots

Data Cube Implementation of Characterization



- Each dimension represents generalized values for one attribute
- A cube cell stores aggregate values, e.g., count, amount
- A "sum" cell stores dimension summation values

A Sample Data Cube

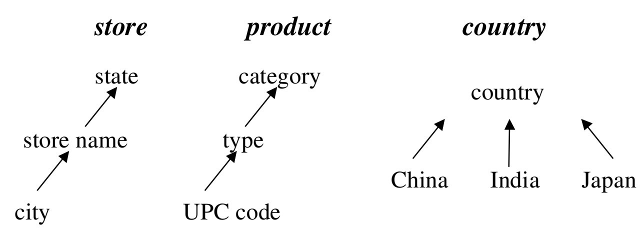


Sample Operations

- Roll up: summarize data
  - *total sales volume last year by product category by region*
- Roll down, drill down, drill through: go from higher level to lower level summary *For the product category, find the detailed sales data for each salesperson by date*
- Slice and dice: select and project
  - *Sales of beverages in the West over last 6 months*
- Pivot: reorient cube

Data Cube

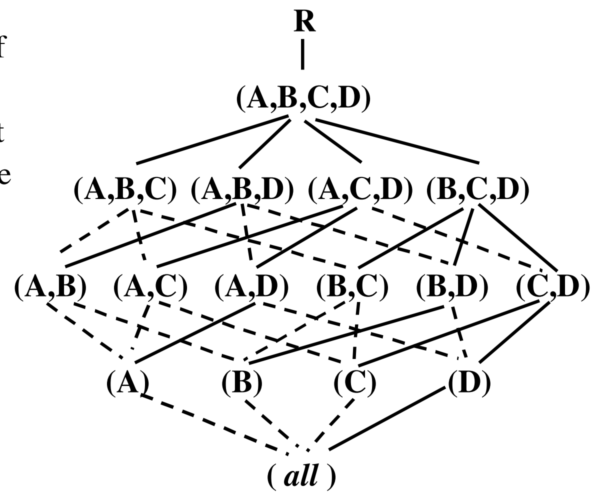
- Popular model for OLAP
- Two kinds of attributes
  - measures (numeric attributes)
  - dimensions



Cuboid Lattice

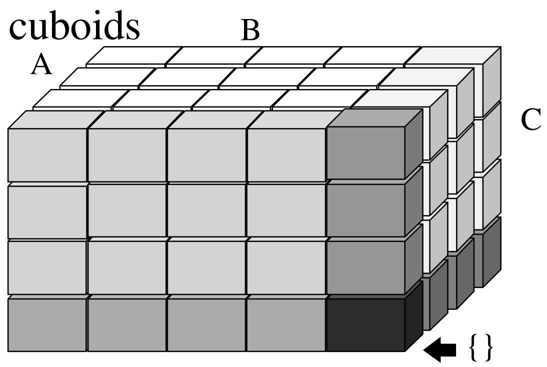
Data cube can be viewed as a lattice of cuboids

- The bottom-most cuboid is the base cube.
- The top most cuboid contains only one cell.



Cube Computation -- Array Based Algorithm

- An MOLAP approach: the base cuboid is stored as multidimensional array.
- Read in a number of cells to compute partial cuboids



{ABC}
{AB}
{AC}
{BC}
{A}
{B}
{C}
{ }

ROLAP versus MOLAP

- ROLAP
  - Exploits services of relational engine
  - Provides additional OLAP services
    - design tools for DSS schema
    - performance analysis tool to pick aggregates to materialize
  - Some SQL queries are hard to formulate and can be time consuming to execute

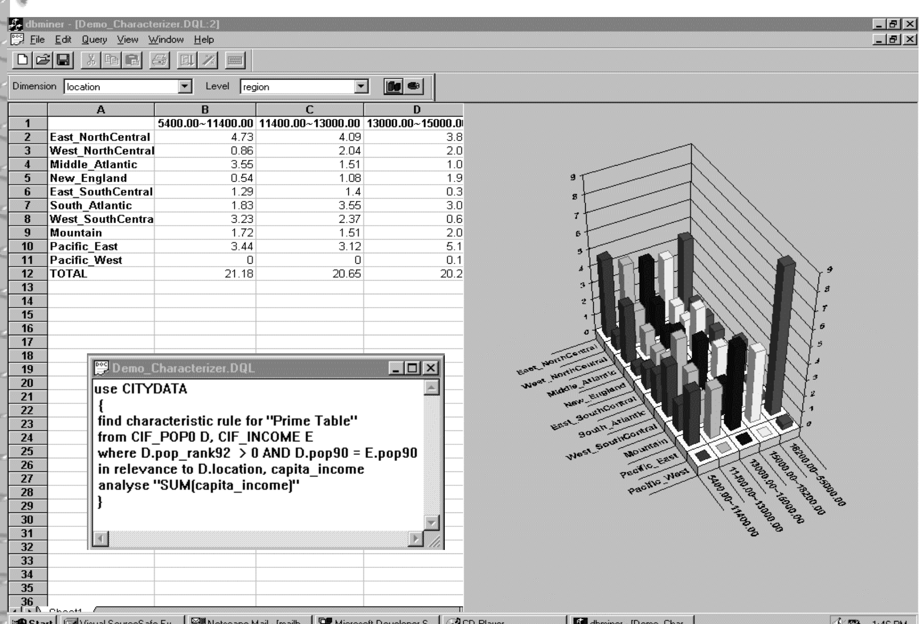
ROLAP versus MOLAP

- MOLAP
  - the storage model is an n-dimensional array
  - Front-end multidimensional queries map to server capabilities in a straightforward way
  - Direct addressing abilities
  - Handling sparse data in array representation is expensive
  - Poor storage utilization when the data is sparse

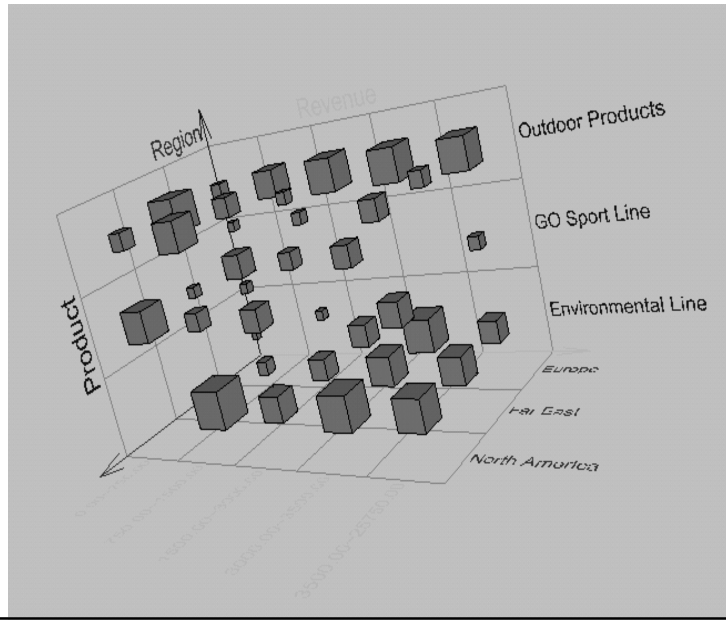
Methodologies of Multiple Level Data Mining

- Progressive generalization (roll-up: easy to implement).
- Progressive deepening (drill-down: conceptually desirable).
  - Start at a rather high level, find strong regularities at such a level
  - Selectively and progressively deepen the knowledge mining process down to deeper levels to find regularities at lower levels.
- Interactive up and down:
  - Roll-up and drill-down to different levels, including setting different thresholds and focuses.
- Implementation: save a "minimally generalized relation".
  - Specialization of a generalized relation: Generalize the minimally generalized relation to appropriate levels.

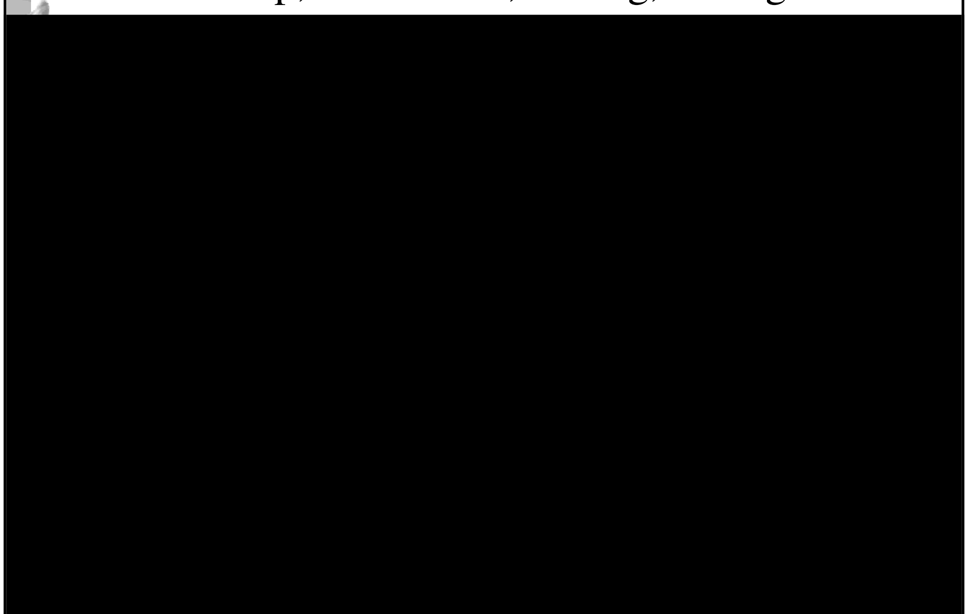
Characterization



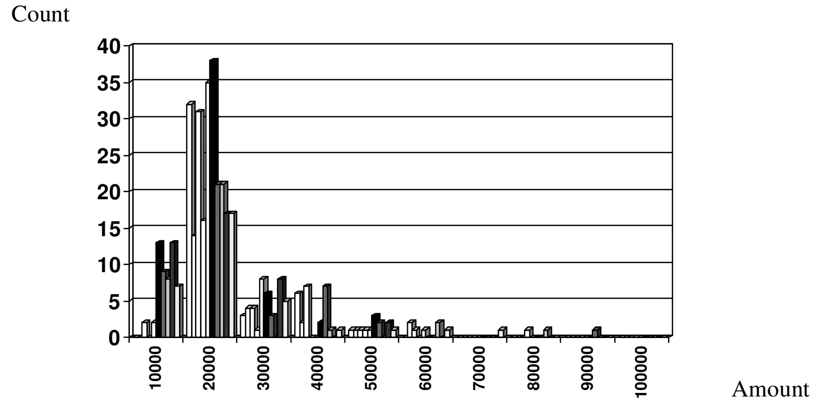
Visualization of a Data Cube

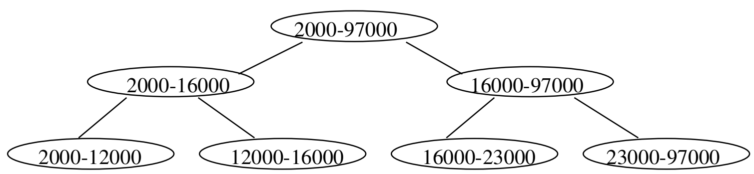


Roll-up, Drill-down, Slicing, Dicing



Automatic Generation of Numeric Hierarchies





Problem: Dates

12

★ Dates are ***very*** important, yet most data mining algorithms do not support them well

★ Solution:

- ➢ Provide well-used measurements in industry, such as Recency and Frequency (of RFM).
- ➢ Provide strong support for date operations (days between dates, day-of-week, etc).

Ronny Kohavi

9

Deviation Analysis in Large Databases

- Major trend and characteristics vs. deviations.
  - E.g., mutual funds which perform much better (or worse) than average
- Data deviation analysis: Discover and describe the set(s) of data which deviate from major trend/characteristics
- A method for trend deviation analysis in large databases
  - A-O induction on time for mining trends at multiple time scales.
  - Generalization or removal of less relevant attributes.
  - Find major and minor trends by data clustering and data distribution analysis
  - Smoothing, similarity matching, and trend analysis

Exploration of OLAP Data Cubes

- Use Data Cube operations to examine data
- Look for surprises (deviation detection)
- Three values of interest
  - SelfExp: Surprise value of specific cell at current level
  - InExp: Degree of surprise beneath this cell (max for all paths)
  - PathExp: Degree of surprise for specific path beneath this cell

InExp Example

“highlight exceptions”

Value indicated by thickness of box

Dimensions: Product, Region, and Time

Product	(All)
Region	(All)

Time

Sum of Sales	Month											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Total		2%	0%	2%	2%	4%	3%	0%	-8%	0%	-3%	4%

Drill down Product

large SelfExp value

large InExp values, drill down along Region

Region	(All)

Avg. Sales	Month											
Product	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Birch-B		10%	-7%	3%	-4%	15%	-12%	-3%	1%	42%	-14%	-10%
Chery-S		1%	1%	4%	3%	5%	5%	-9%	-12%	1%	-5%	5%
Cola		-1%	2%	3%	4%	9%	4%	1%	-11%	-8%	-2%	7%
Cream-S		3%	1%	6%	3%	3%	8%	-3%	-12%	-2%	1%	10%
Diet-B		1%	1%	-1%	2%	1%	2%	0%	-6%	-1%	-4%	2%
Diet-C		3%	2%	5%	2%	4%	7%	-7%	-12%	-2%	-2%	8%
Diet-S		2%	-1%	0%	0%	4%	2%	4%	-9%	5%	-3%	0%
Grape-S		1%	1%	0%	4%	5%	1%	3%	-9%	-1%	-8%	4%
Jolt-C		-1%	-4%	2%	2%	0%	-4%	2%	6%	-2%	0%	0%
Kiwi-S		2%	1%	4%	1%	-1%	3%	-1%	-4%	-4%	0%	1%
Old-B		4%	-1%	0%	1%	5%	2%	7%	-10%	3%	-9%	1%
Orang-S		1%	1%	3%	4%	2%	1%	-1%	-1%	-6%	-4%	0%
Sasprla		-1%	2%	1%	3%	-3%	5%	-10%	-2%	-1%	1%	5%

Drill down Region for a product

Product	Diet-S

Avg.Sales	Month											
Region	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
C		0%	-2%	0%	1%	4%	1%	5%	-6%	2%	-2%	-2%
E		0%	2%	-5%	7%	0%	5%	-40%	10%	-33%	2%	8%
S		0%	-1%	3%	-2%	2%	-2%	19%	-1%	12%	-1%	0%
W		5%	1%	0%	-2%	6%	6%	2%	-17%	9%	-7%	2%

Market	(All)
Product	Cola

Avg.Sales	Month											
Region	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
C		3%	1%	4%	1%	4%	10%	-11%	-14%	-3%	5%	11%
E		-3%	3%	4%	4%	13%	2%	0%	-10%	-13%	-3%	5%
S		2%	-1%	1%	9%	6%	3%	21%	-15%	1%	-5%	4%
W		-2%	2%	2%	4%	12%	1%	1%	-9%	-11%	-4%	6%

Product	Birch-B
Region	E

Sum of Sales	Month											
State	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Massachusetts		0%	-4%	0%	-8%	0%	-2%	0%	7%	35%	-[illegible]4%	-[illegible]6%
New-Hampshire		5%	[illegible]0%	-[illegible]3%	[illegible]7%	3%	[illegible]4%	-27%	-[illegible]7%	57%	-[illegible]%	-4[illegible]%
New-York		[illegible]8%	-[illegible]0%	8%	-4%	24%	-[illegible]9%	-[illegible]%	[illegible]%	44%	-[illegible]5%	-3%

Determining Exceptions

- Consider variation in values for all dimensions of a cell

  <Birch-B, Dec> --- exception
  <Birch-B, Nov> --- not exception

- Find exceptions at all levels of abstraction

  <Birch-B, Oct> --- exception
  No need to mark
  <Birch-B, Oct, Massachusetts>
  <Birch-B, Oct, NewHampshire>
  <Birch-B, Oct, NewYork>

- Maintain computational efficiency

Anticipated and surprising values

- Anticipated value $a_{i_1 i_2 \ldots i_n}$ at dimension dr ($1 \leq r \leq n$) function f of various abstraction levels
- Value $y_{i_1 i_2 \ldots i_n}$ is an "exception" if $s_{i_1 i_2 \ldots i_n} > \tau$ (= 2.5)

$$s_{i_1 i_2 \ldots i_n} = \frac{|y_{i_1 i_2 \ldots i_n} - a_{i_1 i_2 \ldots i_n}|}{\sigma_{i_1 i_2 \ldots i_n}}$$

- f can return
  - sum of arguments
  - product of arguments
- σ is estimated as mean value over all cells raised to power *p* (maximum likelihood principle)

Summarizing exceptions

- SelfExp: $s_{i_1 i_2 \ldots i_n} > \tau$
- InExp: Maximum SelfExp over all cells underneath this cell
- PathExp: Maximum of SelfExp over all cells along one path

Algorithm

- Aggregate computation: Compute means, stddev, etc., for each level of abstraction
- Model fitting: Find model parameters, calculate residuals
- Summarize exceptions: similar to first pass

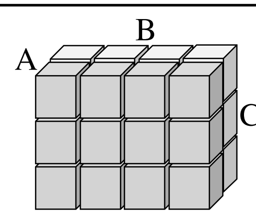
Example

A = 1

B \ C	1	2	3
1	5	2	8
2	7	1	3
3	6	1	3
4	1	9	3

A = 2

B \ C	1	2	3
1	5	2	3
2	7	1	3
3	6	1	3
4	7	2	3



{ABC}
{AB}
{AC}
{BC}
{A}
{B}
{C}
{ }

- Computing averages: {111} = 5, {112} = 2, .., {11*} = 15/3=5, {12*} = 11/3=3.67, {1**} = 45/12=3.67
- Computer coefficients (ElementAvg - CoefParents): c{1**} = 3.67, c{11*} = 5 - ({1**} + {2**}) = -2.25
- Compute model parameters (l-values): l{123} = c{***} + c{1**} + c{*2*} + c{**3} + c{12*} + c{1*3} + c{*23} + c{123}
- Compute exceptions

OLAP Mining (OLAM): An Integration of Data Mining and Data Warehousing

- On-line analytical mining of data warehouse data: integration of mining and OLAP technologies.
- Necessity of mining knowledge and patterns at different levels of abstraction by drilling/rolling, pivoting, slicing/dicing, etc.
- Interactive characterization, comparison, association, classification, clustering, prediction.
- Integration of data mining functions, e.g., first clustering and then association