DATA MINING AND DATA WAREHOUSING

(BCSDM515)

By,
Anusha K S

Assistant Professor, Dept. of CSE
VVCE, Mysuru.

MODULE 2

Data warehouse implementation & Data mining: Efficient Data Cube computation: An overview, Indexing OLAP Data: Bitmap index and join index, Efficient processing of OLAP Queries.

Introduction: What is Data Mining? Motivating Challenges, The Origins of Data Mining, Data Mining Tasks. Data: Types of Data, Data Quality, Data Preprocessing.

SLT: OLAP server Architecture ROLAP versus MOLAP Versus HOLAP.

Data warehouse implementation

1. **Efficient Data Cube computation**

    I. Compute Data
    II. Materialization

2. **Indexing OLAP DATA**

    I. Bitmap Indexing
    II. Join Indexing

3. **Efficient processing of OLAP Queries**

Efficient Data Cube computation: An overview

- Multidimensional data analysis is the efficient computation of aggregations across many sets of dimensions

- A data cube is a lattice of cuboids

- Suppose that you want to create a data cube for AllElectronics sales that contains the following: city, item, year, and sales_in_dollars. You want to be able to analyze the data, with queries such as the following:

  - "Compute the sum of sales, grouping by city and item"
  - "Compute the sum of sales, grouping by city"
  - "Compute the sum of sales, grouping by item"

- "Compute Cube" operator computes aggregates over all subsets of the dimensions specified in the operation.

- The sql syntax could be defined as
  Define cube sales_cube [city, year, item] : sum(sales_in_dollars)
  "**compute cube** sales_cube"

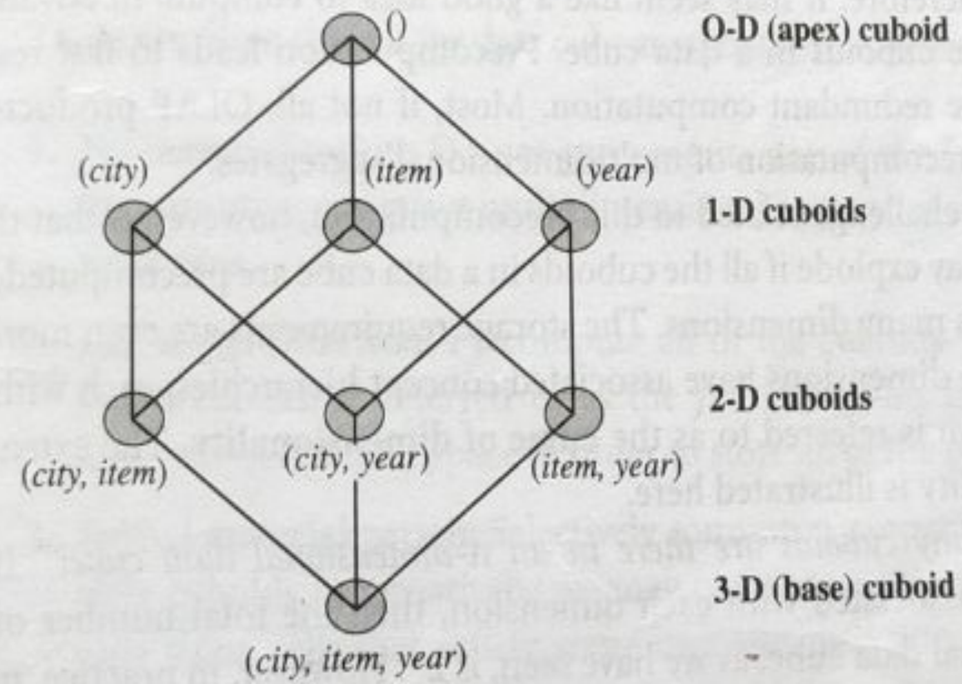


Figure 4.14 Lattice of cuboids, making up a 3-D data cube. Each cuboid represents a different group-by. The base cuboid contains *city*, *item*, and *year* dimensions.

"How many cuboids are there in an n*-dimensional data cube?"*

- the total number of cuboids for an *n*-dimensional data cube, is **2^*n***

Curse of Dimensionality

- The storage requirements are more excessive when many of dimensions have associated **concept hierarchies**. This problem is referred as “Curse of Dimensionality”

- Example: time is usually explored not at only one conceptual level (year : day<month<quarter<year)

$$Total\ number\ of\ cuboids = \prod_{i=1}^{n} (L_i + 1),$$

Where,

Li is the number of levels associated with dimension *i*.

One is added to *Li in equation* to include the *virtual* top level

Three choices for data cube materialization

1. **No materialization:** Do not precompute any of the "non-base" cuboids. This leads to computing expensive multidimensional aggregates on-the-fly, which can be extremely slow.

2. **Full materialization:** Precompute all of the cuboids. The resulting lattice of computed cuboids is referred to as the *full cube*. This choice typically requires huge amounts of memory space in order to store all of the precomputed cuboids.

3. **Partial materialization:** Selectively compute a proper subset of the whole set of possible cuboids. Alternatively, we may compute a subset of the cube, which contains only those cells that satisfy some user-specified criterion, such as where the tuple count of each cell is above some threshold. Partial materialization represents an interesting trade-off between storage space and response time.
   - Partial materialization of cuboids or sub-cubes should consider 3 factors
     Identify, exploit, update

Indexing OLAP DATA

- To facilitate efficient data accessing, most data warehouse systems support index structures and materialized views (using cuboids).

1. Bitmap Indexing

- The bitmap indexing method is popular in OLAP products because it allows quick searching in data cubes.

- The bitmap index is an **alternative representation of the record_ID** (RID) list.

- In the bitmap index for a given attribute, there is a distinct bit vector, Bv, for each value v in the attribute's domain. If a given attribute's domain consists of n values, then n bits are needed for each entry in the bitmap index (i.e., there are n bit vectors).

- If the attribute has the value v for a given row in the data table, then the bit representing that value is set to 1 in the corresponding row of the bitmap index. All other bits for that row are set to 0.

Base table

RID	*item*	*city*
R1	H	V
R2	C	V
R3	P	V
R4	S	V
R5	H	T
R6	C	T
R7	P	T
R8	S	T

item bitmap index table

RID	H	C	P	S
R1	1	0	0	0
R2	0	1	0	0
R3	0	0	1	0
R4	0	0	0	1
R5	1	0	0	0
R6	0	1	0	0
R7	0	0	1	0
R8	0	0	0	1

city bitmap index table

RID	V	T
R1	1	0
R2	1	0
R3	1	0
R4	1	0
R5	0	1
R6	0	1
R7	0	1
R8	0	1

Note: H for "home entertainment," C for "computer," P for "phone," S for "security," V for "Vancouver," T for "Toronto."

Figure 4.15 Indexing OLAP data using bitmap indices.

2. Join Indexing

- join indexing registers the joinable rows of two relations from a relational database.

- For example, if two relations R(RID, A) and S(B, SID) join on the attributes A and B, then the join index record contains the pair (RID, SID), where RID and SID are record identifiers from the R and S relations, respectively.

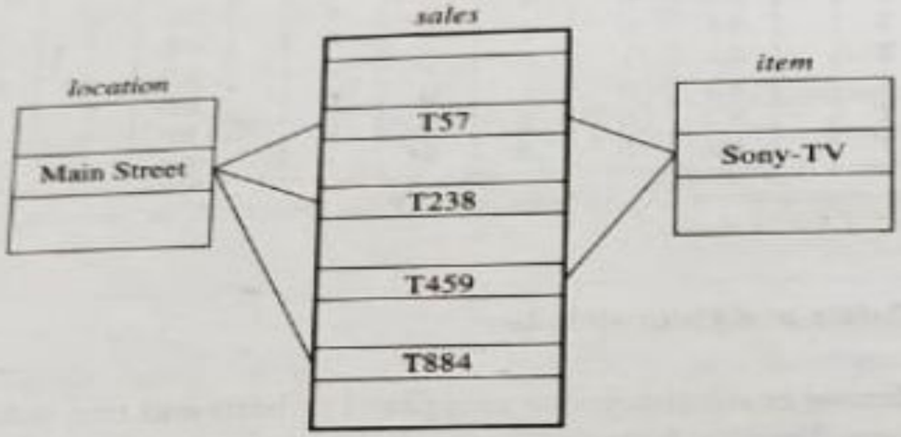


Figure 4.16 Linkages between a *sales* fact table and *location* and *item* dimension tables.

Join index table for *location/sales*

location	*sales_key*
...	...
Main Street	T57
Main Street	T238
Main Street	T884
...	...

Join index table for *item/sales*

item	*sales_key*
...	...
Sony-TV	T57
Sony-TV	T459
...	...

Join index table linking *location* and *item* to *sales*

location	*item*	*sales_key*
...	...	...
Main Street	Sony-TV	T57
...	...	...

Figure 4.17 Join index tables based on the linkages between the *sales* fact table and the *location* and *item* dimension tables shown in Figure 4.16.

Efficient processing of OLAP Queries

- The purpose of materializing cuboids and constructing OLAP index structures is to speed up query processing in data cubes
- The steps used for query processing in Data Cube should proceed as follows

1. Determine which operations should be performed on the available cuboids
2. Determine to which materialized cuboid(s) the relevant operations should be applied

OLAP query processing. Suppose that we define a data cube for *AllElectronics* of the form "*sales_cube* [*time, item, location*]: sum(*sales_in_dollars*)." The dimension hierarchies used are "*day* < *month* < *quarter* < *year*" for *time*; "*item_name* < *brand* < *type*" for *item*; and "*street* < *city* < *province_or_state* < *country*" for *location*.

Suppose that the query to be processed is on {*brand, province_or_state*}, with the selection constant "*year = 2010*." Also, suppose that there are four materialized cuboids available, as follows:

- cuboid 1: {*year, item_name, city*}
- cuboid 2: {*year, brand, country*}
- cuboid 3: {*year, brand, province_or_state*}
- cuboid 4: {*item_name, province_or_state*}, where *year* = 2010

OLAP Server Architectures

ROLAP versus MOLAP versus HOLAP

The Cube in a data warehouse are stored in three different modes

- Multidimensional OLAP (MOLAP) servers
- Relational OLAP (ROLAP) servers
- Hybrid OLAP (HOLAP) servers

Multidimensional OLAP

Sr.No	MOLAP	ROLAP
1	Information retrieval is fast	Information retrieval is comparatively slow
2	Uses sparse array to store data-sets	Uses relational table
3	MOLAP is best suited for inexperienced users, since it is very easy to use.	ROLAP is best suited for experienced users
4	Maintains a separate database for data cubes.	It may not require space other than available in the Data warehouse.
	DBMS facility is weak	DBMS facility is strong

DATA MINING

What is data mining, Challenges, Data Mining Tasks, Data: Types of Data, Data Quality, Data Preprocessing.

- Datamining is a technology that blends traditional data analysis method with sophisticated algorithms for processing large volume of data

- Data mining is a process of automatically discovering useful information in large data repositories.

- PURPOSE !

- WHY !

- WHERE !

- TOOLS : Rapidminer, orange, Teradata, IBM, Weka, Oracale data Mining.

DATA MINING AND KNOWLEDGDE DISCOVERY

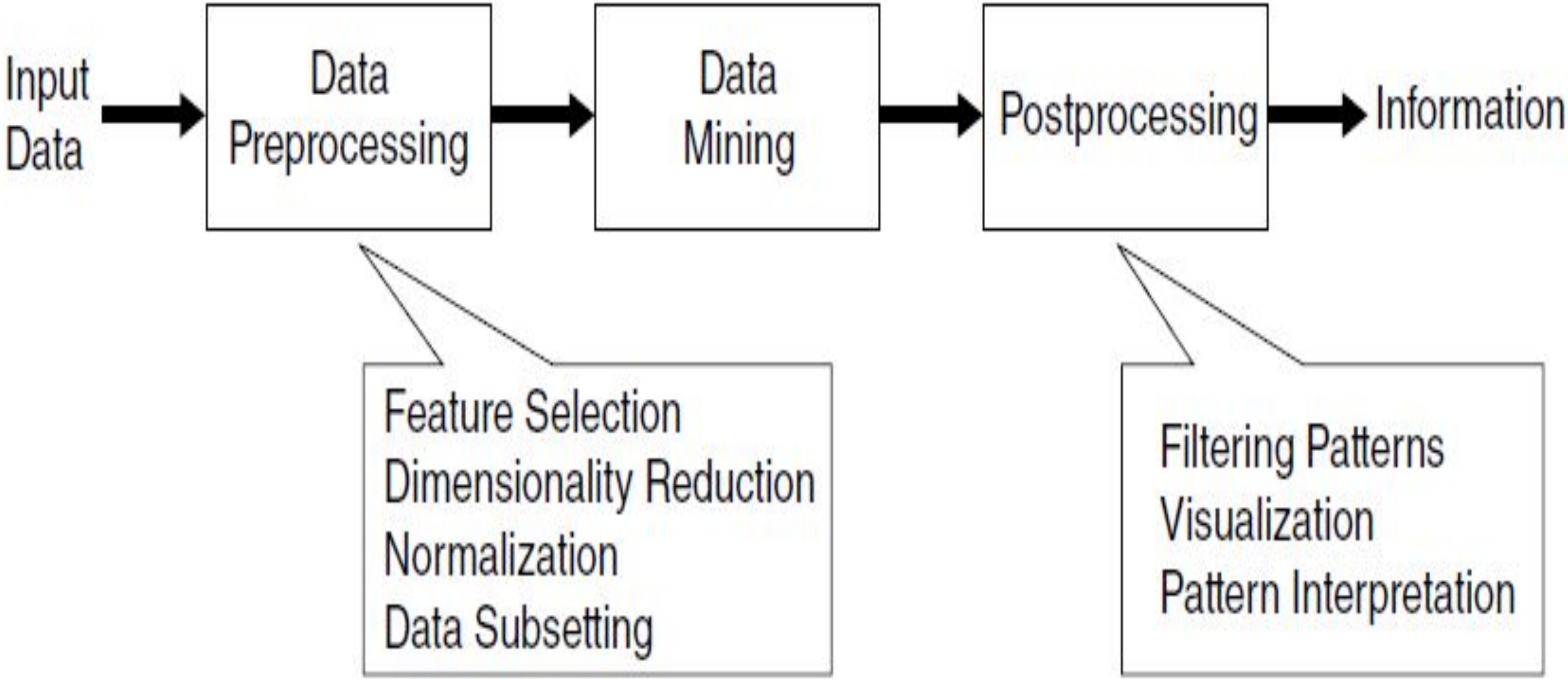


Figure 1.1. The process of knowledge discovery in databases (KDD).

Challenges

- Scalability
- High Dimensionality
- Heterogeneous and Complex Data
- Data Ownership and Distribution
- Non-traditional Analysis

DATA MINING TASKS

- **PREDICTIVE TASKS**

The objective of these tasks is to predict the values of a particular attributes based on the values of other attributes.

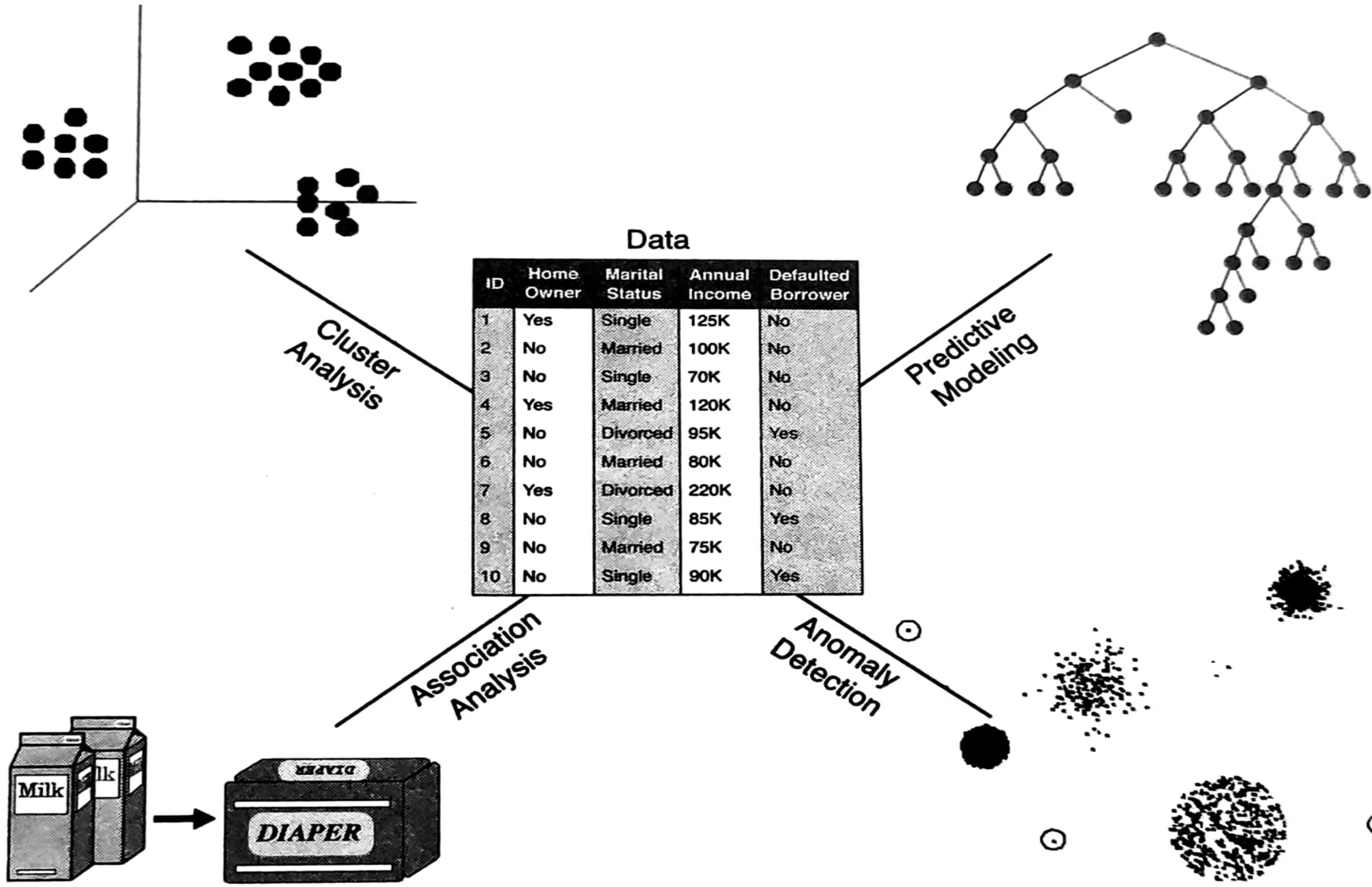
Example: A medical practioner trying to diagnose the disease based on the medical test results.

- **DESCRIPTIVE TASKS**

The objective is to derive patterns (correlations, trends, clusters and anomalies) that summarize the underlying relationships in data.

Example: A retailer trying to identify products that are purchased together

FOUR CORE DATA MINING TASKS



ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	80K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Figure 1.3. Four of the core data mining tasks.

TYPES OF DATA

- Qualitative data

Arise when the observations fall into separate distinct categories (discrete)

- Quantitative data (numeric)

Arise when the observations are “counts” or “measurements” (discrete or continuous)

ATTRIBUTE AND MEASUREMENT OF DATA

- Attribute

- Measurement scale

DIFFERENT TYPES OF ATTRIBUTES

- **Distinctness** (Nominal (= & !=)) – info to distinguish one object from another
  Ex: zip codes, employee ID

- **order** (ordinal (<, <=, >, >=)) – provide enough info to order objects
  Ex: Grades, street numbers

- **Interval** (Addition (- & +)) – The differences between values are meaningful
  Ex: calendar dates

- **Multiplication** (ratio (* & /)) – Both difference and ratio are meaningful
  Ex: counts, age

General Characteristics Of Data Sets

- Dimensionality
- Sparsity
- Resolution

Types Of Data Sets

- **Record Data** : Transaction data, data matrix, document-term Metrix
- **Graph-Based Data** : Data with relationship among objects, Data with objects that are graph
- **Ordered Data** : Sequential data, time series data, spatial data

DATA QUALITY

Measurement and Data Collection Errors:

1. Noise
2. Artifacts – data flow
3. Precision – is a closeness of repeated measurements to one another
4. Bias – is a systematic variation of measurements from the quality being measured
5. Accuracy

DATA QUALITY ISSUES

1. Outliers
2. Missing Values
3. Inconsistent Values
4. Duplicate Data

DATA PREPROCESSING

- Aggregation
- Sampling
- Dimensionality reduction
- Feature subset selection: Embedded, filter and Wrapper
- Feature Creation
- Discretization and Binarization
- Variable/Attribute Transformation

Thank you