

Databases Illuminated

Chapter 15

Data Warehouses and Data Mining

Intro to Data Warehouses

- Term coined by W.H. Inmon
 - “a subject-oriented, integrated, non-volatile, time-varying collection of data that is used primarily in organizational decision making”
- Enterprises use historical and current data taken from operational databases as resource for decision making
- Data warehouses store massive amounts of data
- Typical uses
 - decision support systems (DSS)
 - on-line analytical processing (OLAP)
 - data mining
- Major DB vendors provide warehouse features, including analytical tools
- SQL:1999 includes data mining functions

Characteristics of Operational Databases

- Support online transaction processing (OLTP)
 - use limited number of repetitive transactions
 - transactions involve a few tuples at a time
- Serve the information needs of end users
- Support day-to-day business operations
- Require high availability and efficient performance
- Handle large volume of transactions
- Must deliver query responses quickly
- Must do updates quickly
- State must reflect current environment of the enterprise

Characteristics of Data Warehouses

- Support **on-line analytical processing- OLAP**
 - Examine large amounts of data to produce results
 - Allow complex queries, often using grouping
 - Support time-series analysis using historical data
- Used for decision making
- Contain very large amount of data
- Have data from multiple operational databases, taken at different periods of time (historical data)
- Sources may have different models or standards; data warehouse integrates the data
- May include data from other sources, summarized data, metadata
- Optimized for efficient query processing and presentation of results
- Updates done periodically; not in real time
- Support **data mining**-discovering new information by searching large amounts of data
 - purpose is to discover patterns or trends in the data

Data Warehouse Architecture-1

- See **Figure 15.1**
- Must support ad-hoc queries and unanticipated types of analysis
- Input data
 - Taken from various data sources
 - Multiple operational databases
 - Independent files
 - Environmental data-e.g. geographical or financial data
 - Extracted using back-end system tools-accommodate differences among heterogeneous sources
 - Reformatted into a consistent form
 - Checked for integrity and validity- **data cleaning**
 - Put into the data model for the warehouse
 - Loaded - long transaction due to large volume

Data Warehouse Architecture-2

- DBMS for data warehouse has
 - System catalog that stores metadata
 - Other standard database system components
- **Data marts** - segments of the data organized into subsets that focus on specific subjects; e.g. may contain specialized information about a single department
- Data warehouse output
 - Supports queries for OLAP
 - Provides information for decision support systems
 - Provides data for data mining tools
 - Can result in new knowledge, which can then be used as a data source

Data Refresh

- Data from all sources must be refreshed periodically
- New data is added to the existing warehouse, if there is room; old data is kept as long as it is useful
- Data no longer used is purged periodically
- Frequency and scope of updates depends on the environment
- Factors for deciding the update policy
 - How much storage is available
 - Whether the warehouse needs recent data
 - Whether warehouse can be off-line during refresh
 - How long the process of transmitting the data, cleaning, formatting, loading, and building indexes will take
- Usual policy is to do a partial refresh periodically

Data Models for Data Warehouses

- Generally use a multidimensional model
- **Data cube** - multidimensional matrix for storing data
 - Can view the data by dimension of interest
 - Possible operations on data cube
 - **pivoting** - rotating to display a different dimension
 - **rollup** - displaying a coarser level of data granularity, by combining or aggregating data
 - **drill-down** - showing more detail on some dimension, using finer granularity for the data; requires that the more detailed data be available
 - **slicing** - examining a portion of the data cube using a selection with equality conditions for one or more dimensions; appears as if the user has cut through the cube in the selected directions
 - **dicing**- specifying a range of values in a selection
 - **Cross-tabulation** – displaying totals for the rows and columns in a two-dimensional spreadsheet-style display
- **Hypercube** - data cube of dimension > 3
 - Possible to do pivoting, rollup, drilling down, slicing, dicing
 - No physical representation of cube itself

Schemas for Data Warehouses

- **Multidimensional OLAP (MOLAP)** systems use multidimensional arrays
- **Relational OLAP (ROLAP)** systems use relational model
- **ROLAP warehouse** has multiple relational tables
- **Star schema**
 - Central **fact table** of un-aggregated, observed data
 - Has attributes that represent dimensions, plus dependent attributes
 - Each dimension has its own dimension table
 - Dimension tables have corresponding dimension attributes in fact table, usually foreign keys there
- **Snowflake schema**
 - Variation in which normalized dimension tables have dimensions themselves
- See **Figure 15.4**

Warehouse Queries in SQL92 Form

- SQL92 aggregate functions SUM, COUNT, MAX, MIN and AVG allow some slicing and dicing queries. Form is

```
SELECT <grouping attributes> <aggregation function>  
FROM <fact table> JOIN <dimension table(s)>  
WHERE <attribute = constant>... <attribute = constant>  
GROUP BY <grouping attributes>;
```

SQL:1999 Warehouse Queries

- SQL:1999 includes functions for
 - **stddev** (standard deviation) and **variance** for single attributes – measures of data spread from mean
 - **correlation** and **regression**, which apply to pairs of attributes
 - **rank** for data values
 - GROUP BY extended with CUBE and ROLLUP options

Indexes for Warehouses

- Efficient indexes important because of large quantity of data
- Allow queries to be executed in reasonable time
- Since data is relatively static, cost of maintaining indexes is not a factor
- Special indexing techniques used for warehouses
 - bitmap indexing
 - join indexing

Bitmap Indexes

- Can be constructed for any attributes that have a limited number of distinct possible values-small domain
- For each value in the domain, a bit vector is constructed to represent that value, by placing a 1 in the position for that value
- Take much less space than standard indexes
- Allow processing of some queries directly from the index

Join Indexes

- Join is slow when tables are large
- Join indexes speed up join queries
- Most join operations are done on foreign keys
- For a star schema, the join operation involves comparing the fact table with dimension tables
- **Join index** relates the values of a dimension table to the rows of the fact table
- For each value of the indexed attribute in the dimension table, join index stores the tuple IDs of all the tuples in the fact table having that value

Views and Query Modification

- Views are important in data warehouses for customizing the user's environment
- SQL operators, including CUBE and ROLLUP, can be performed on views as well as on base tables
- SQL CREATE VIEW command defines the view, but does not create any new tables
- Can execute a query for a view by **query modification**, replacing the reference in the WHERE line by the view definition
- Query modification may be too slow in a warehouse environment

View Materialization

- View materialization – pre-computing views from the definition and storing them for later use
- Indexes can be created for the materialized views, to speed processing of view queries
- Designer must decide which views to materialize; weighs storage constraints against benefit of speeding up important queries

Materialized View Maintenance

- When the underlying base tables change, view should also be updated
- **Immediate view maintenance**, done as part of the update transaction for the base tables; slows down the refresh transaction for the data warehouse
- Alternative is **deferred view maintenance**. Possible policies
 - **Lazy refresh**, update the view when a query using the view is executed and the current materialized version is obsolete
 - **Periodic refresh**, update the view at regular time intervals
 - **Forced refresh**, update the view after a specified number of updates to the underlying base tables
- Process can be done by re-computing the entire materialized view
- For complex views especially with joins or aggregations, may be done incrementally, incorporating only changes to the underlying tables

Data Mining

- Discovering new information from very large data sets
- Knowledge discovered is usually in the form of patterns or rules
- Uses techniques from statistics and artificial intelligence
- Need a large database or a data warehouse

Data Formats for Data Mining

- Data mining application should be considered in the original design of the warehouse
- Operations used in data mining differ from the analytical ones for OLAP and decision support systems
- Requires summarized data as well as raw data taken from original data sources
- Requires knowledge of the domain and of the data mining process
- Best data format is a “flat file” in which all the data for each case of observed values appears as a single record
- If “flat file” not used, data must be prepared and reformatted for data mining

Purpose of Data Mining

- Usually the ultimate purpose is to provide knowledge that will give a company a competitive advantage, enabling it to earn a greater profit
- Goals of data mining
 - Predict the future behavior of attributes
 - Classify items, placing them in the proper categories
 - Identify the existence of an activity or an event
 - Optimize the use of the organization's resources

Types of Knowledge Discovered

- Data mining uses **induction**
- Examines a large number of cases and concludes that a pattern or a rule exists
- Knowledge can be represented as rules, decision trees, neural networks, or frames

Possible Output: Association and Classification Rules

- **Association rules** have form $\{x\} \rightarrow \{y\}$, where x and y are events that occur at the same time.
 - Have measures of **support** and **confidence**.
 - Support is the percentage of transactions that contain all items included in both left and right hand sides
 - Confidence is how often the rule proves to be true; where the left hand side of the implication is present, percentage of those in which the right hand side is present as well
- **Classification rules**, placing instances into the correct one of several possible categories
 - Developed using a **training set**, past instances for which the correct classification is known
 - System develops a method for correctly classifying a new item whose class is currently unknown

Possible Output: Sequential Patterns

- **Sequential patterns** e.g. prediction that a customer who buys a particular product in one transaction will purchase a related product in a later transaction
 - Can involve a set of products
 - Patterns are represented as sequences $\{S1\}$, $\{S2\}$
 - First subsequence $\{S1\}$ is a **predictor** of the second subsequence $\{S2\}$
 - **Support** is the percentage of times such a sequence occurs in the set of transactions
 - **Confidence** is the probability that when $\{S1\}$ occurs, $\{S2\}$ will occur on a subsequent transaction - can calculate from observed data

Time Series Patterns

- A **time series** is a sequence of events that are all of the same type
- Sales figures, stock prices, interest rates, inflation rates, and many other quantities can be analyzed using time series
- Time series data can be studied to discover patterns and sequences
- For example, we can look at the data to find the longest period when the figures continued to rise each month, or find the steepest decline from one month to the next

Data Mining Methods

- **Decision tree**, a method of developing classification rules
- Developed by examining past data to determine how significant attributes and values are related to outcomes
 - Nodes of the tree represent **partitioning attributes**, which allow the set of training instances to be partitioned into disjoint classes
 - The **partitioning conditions** are shown on the branches
- Tree is then used to classify new cases
- See **Figure 15.6**

Regression

- A statistical method for predicting the value of an attribute, Y, (the dependent variable), given the values of attributes X1, X2, ..., Xn (the independent variables)
- Statistical packages allow users to identify potential factors for predicting the value of the dependent variable
- Using **linear regression**, the package finds the contribution or weight of each independent variable, as coefficients, a0, a1, ..., an for a linear function
$$Y = a_0 + a_1 X_1 + a_2 X_2 + \dots + a_n X_n$$
- Formula represents a curve that fits the observed values as closely as possible.
- In data mining, system itself may be asked to identify the independent variables, as well as to find the regression function
- Can also use **non-linear regression**, using **curve-fitting**, finding the equation of the curve that fits the observed values

Neural Networks

- Methods from AI using a set of samples to find the strongest relationships between variables and observations
- Network given training set that provides facts about input values
- Use a learning method, adapting as they learn new information from additional samples
- Hidden layers developed by the system as it examines cases, using generalized regression technique
- System refines its hidden layers until it has learned to predict correctly a certain percentage of the time; then test cases are provided to evaluate it
- One problem: **overfitting** the curve - prediction function fits the training set values too perfectly, even ones that are incorrect (data noise); prediction function will then perform poorly on new data
- Knowledge of how the system makes its predictions is in the hidden layers: users do not see the reasoning; weights assigned to the factors cannot be interpreted in a natural way
- Output may be difficult to understand and interpret

Clustering

- Methods used to place cases into clusters or groups that can be disjoint or overlapping
- Using a training set, system identifies a set of clusters into which the tuples of the database can be grouped
- Tuples in each cluster are similar, and they are dissimilar to tuples in other clusters
- Similarity is measured by using a **distance function** defined for the data

Applications of Data Mining

- **Retailing**
 - Customer relations management (CRM)
 - Advertising campaign management
- **Banking and Finance**
 - Credit scoring
 - Fraud detection and prevention
- **Manufacturing**
 - Optimizing use of resources
 - Manufacturing process optimization
 - Product design
- **Medicine**
 - Determining effectiveness of treatments
 - Analyzing effects of drugs
 - Finding relationships between patient care and outcomes