Database Development and Design (CPT201)

Lecture 13a: Data Mining 1 – Classification, OLAP and Decision Tree

Dr. Wei Wang

Department of Computing

Learning Outcome

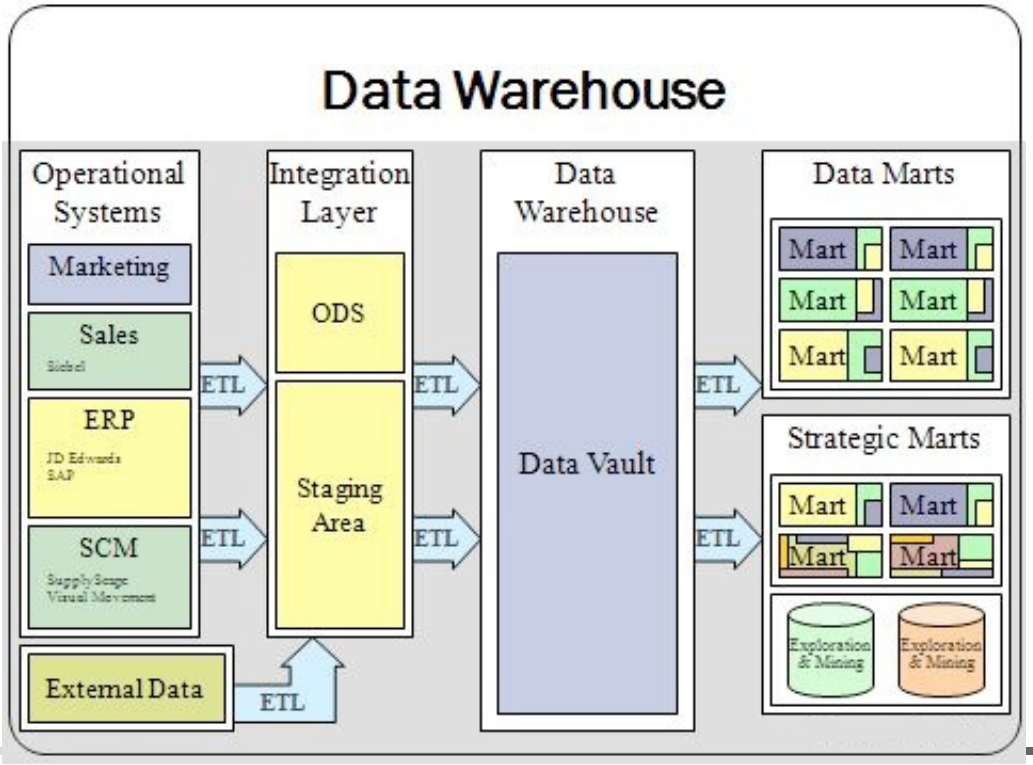
- Data warehouse
- OLAP: Online Analytical Processing
- Classification and Decision Trees

Introduction

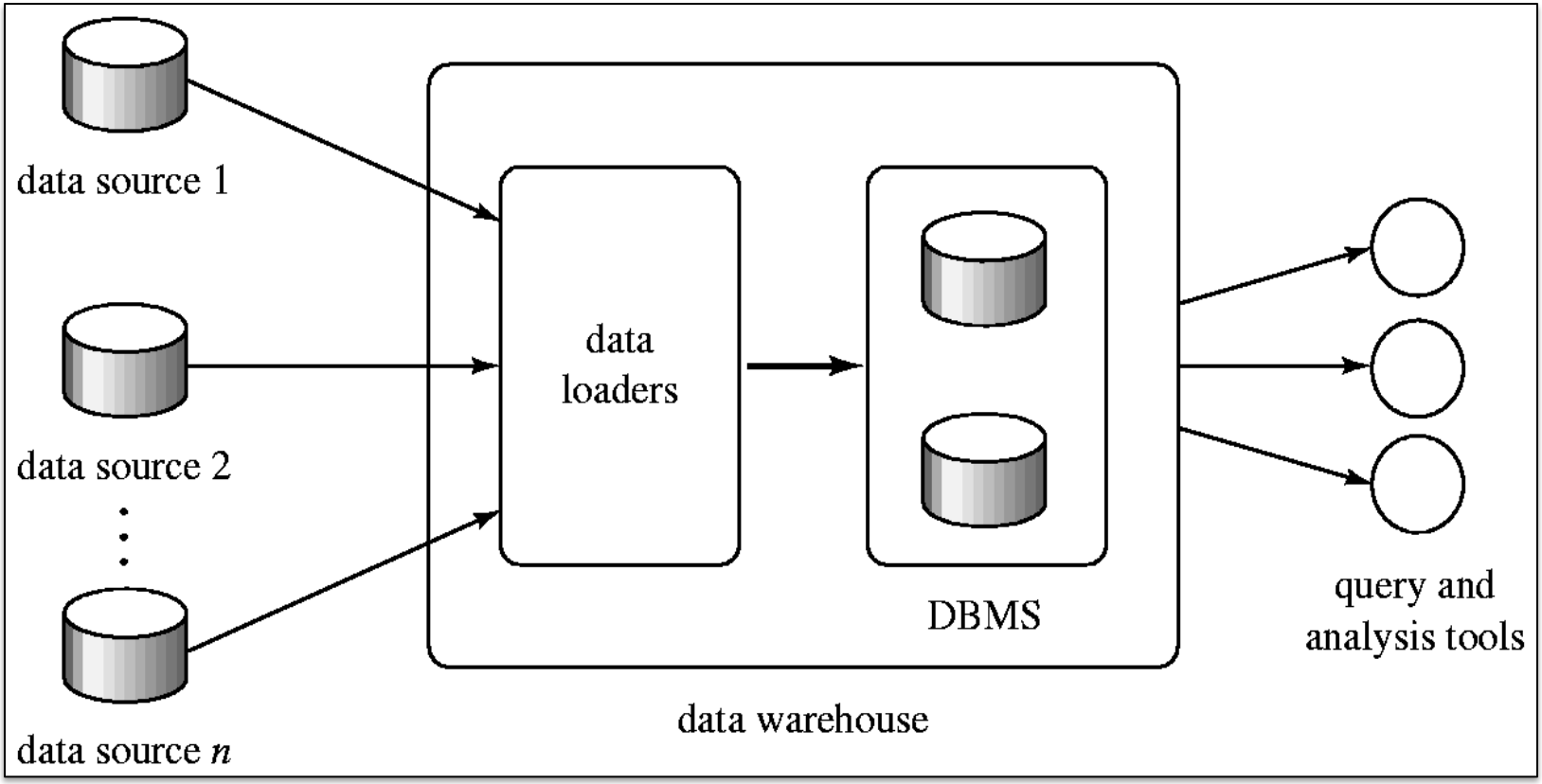
- Increasingly, organisations are analysing current and historical data to identify useful patterns and support business strategies.
- Emphasis is on complex, interactive, exploratory analysis of very large datasets created by integrating data from across all parts of an enterprise.
- Three main trends:
  - Data Warehousing: Consolidate data from many sources in one large repository, e.g., loading, periodic synchronisation of replicas, semantic integration.
  - OLAP, e.g., complex SQL queries and views, queries based on spreadsheet-style operations and multidimensional" view of data, interactive and "online" queries.
  - Data Mining: exploratory search for interesting trends and anomalies.

Data Warehousing

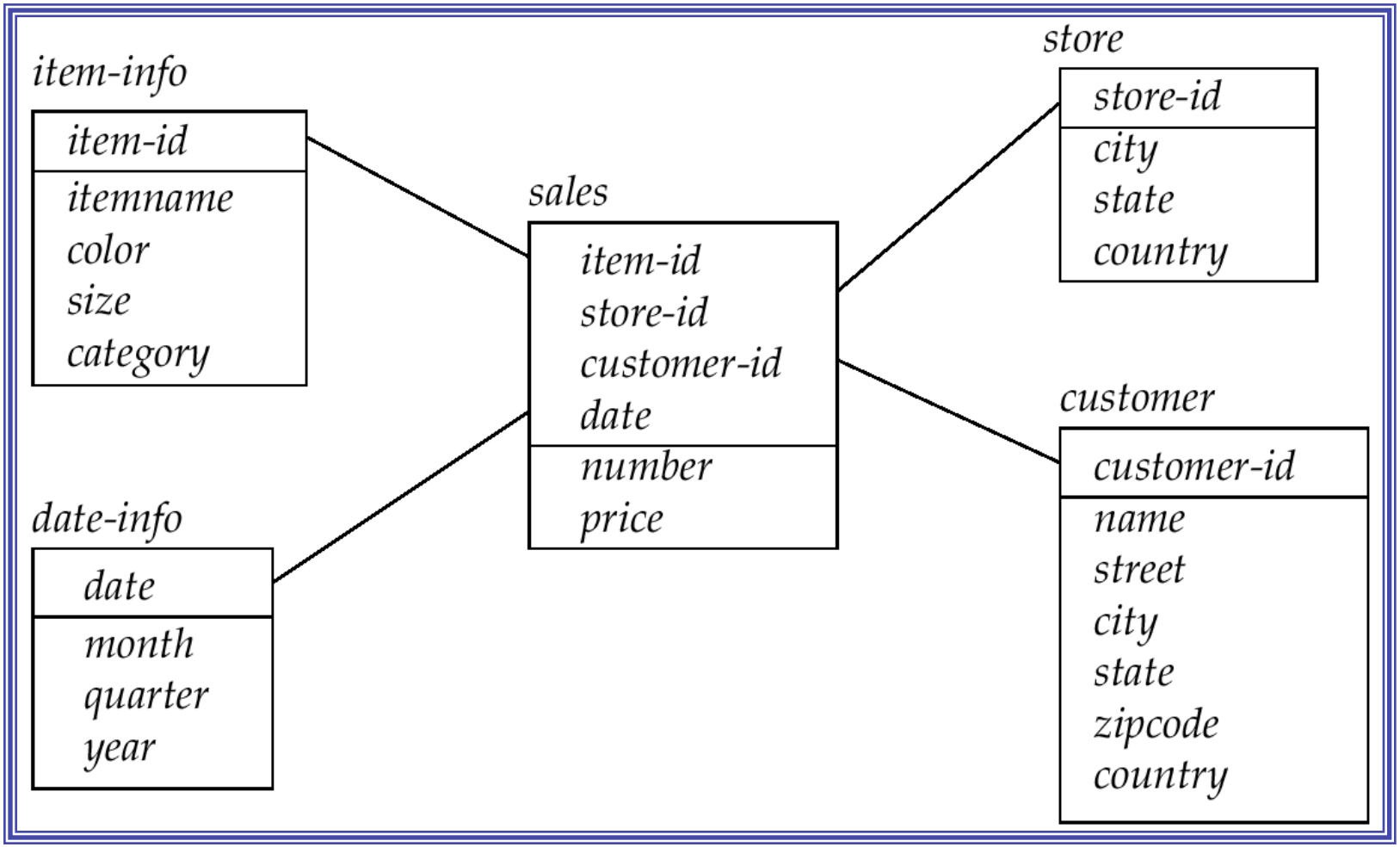
- Integrated data spanning long time periods, often augmented with summary information (ETL: Extract-Transform-Load).
- Several gigabytes to terabytes common.
- Interactive response times expected for complex queries.
- Ad-hoc updates uncommon.



Data Warehouse Architecture



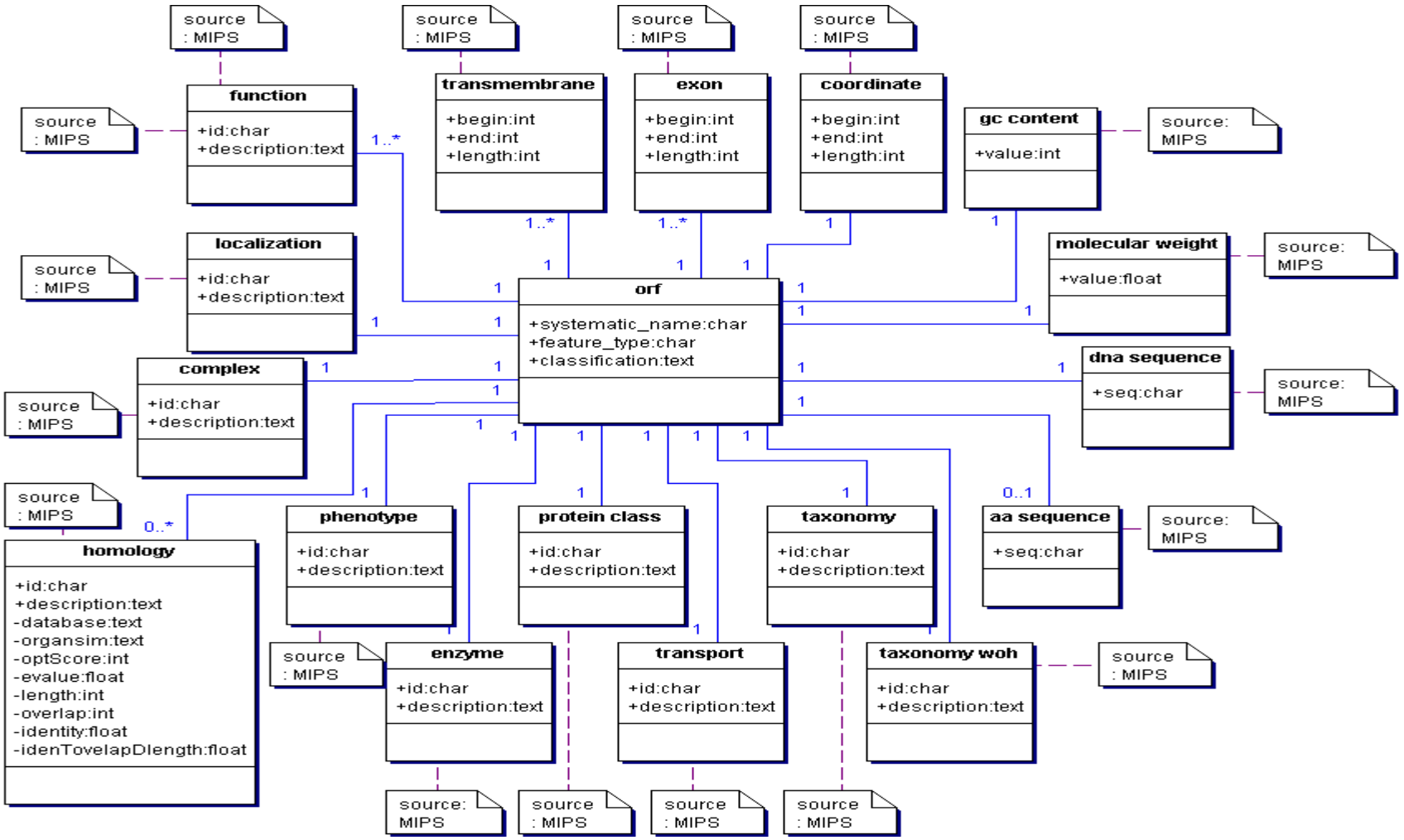
Data Warehouse Schema Example



BLID (Bio-Logical Intelligent Database)

- BLID (Bio-Logical Intelligent Database) is a bioinformatics system designed to help biologists extract new knowledge from raw genome data by providing high-level facilities for both data browsing and analysis.
- Providing intelligent queries and improving scalability.
  - The Yeast genome is the most well-characterised eukaryotic genome and one of the simplest in terms of identifying open reading frames (ORFs).
  - Over 6,000 ORFs, yet about 40% ORFs have no known function.
  - Existing Yeast Databases (KEGG, MIPS, SGD etc.) only support data queries, not knowledge queries.
  - Most existing KDD methods working with dataset, poor scalability.

Example: BLID Relational schema



Warehousing Issues

- Heterogeneous Sources: may access data from a variety of sources and repositories.
- Semantic Integration: when getting data from multiple sources, must eliminate mismatches, e.g., different currencies, schemas.
- Load, Refresh, Purge: must load data, periodically refresh it, and purge too-old data.
- Metadata Management: must keep track of source, loading time, and other information for all data in the warehouse.

Data Analysis and OLAP

- **Online Analytical Processing (OLAP)**
  - Interactive analysis of data, allowing data to be summarised and viewed in different ways in an online fashion (with negligible delay)
- Data that can be modeled as dimension attributes and measure attributes are called **multidimensional data**.
  - As a example, given *sales* relation on clothes shop

    sales(item-name, color, size, number)
  - **Measure attributes**
    - measure some value
    - can be aggregated upon
    - e.g., the attribute *number* of the *sales* relation
  - **Dimension attributes**
    - define the dimensions on which measure attributes (or aggregates thereof) are viewed
    - e.g., the attributes *item_name, color,* and *size* of the *sales* relation

Cross Tabulation of ***sales*** by ***item-name*** and ***color***

size: **all**

item-name \ *color*	dark	pastel	white	Total
skirt	8	35	10	53
dress	20	10	5	35
shirt	14	7	28	49
pant	20	2	5	27
Total	62	54	48	164

- The table above is an example of a cross-tabulation (cross-tab), also referred to as a pivot-table.
  - Values for one of the dimension attributes form the row headers
  - Values for another dimension attribute form the column headers
  - Other dimension attributes are listed on top
  - Values in individual cells are (aggregates of) the values of the dimension attributes that specify the cell.

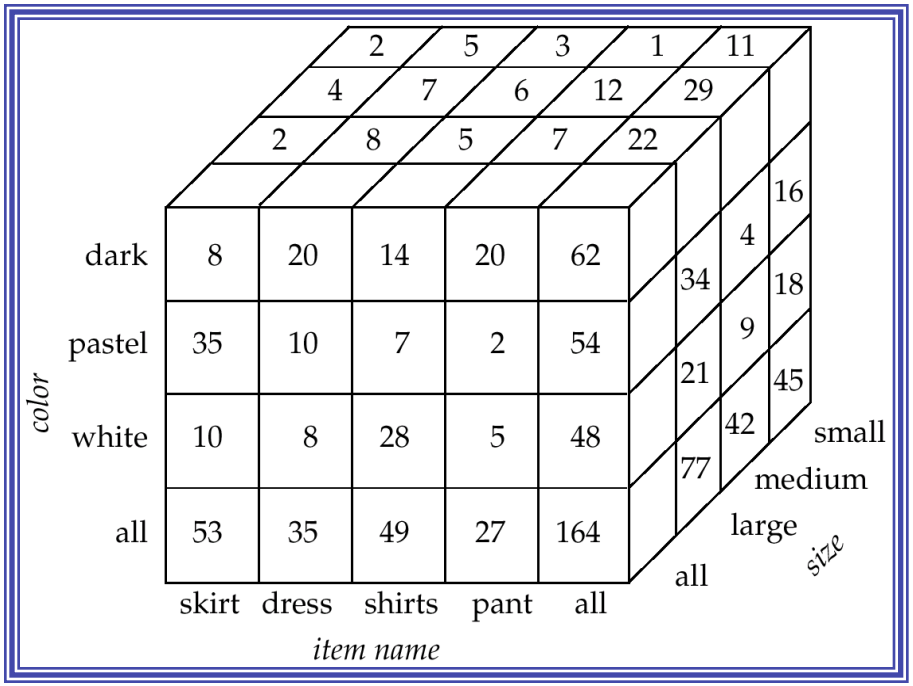
Relational Representation of Cross-tabs

- Cross-tabs can be represented as relations
- We use the value **all** to represent aggregates
  - The SQL:1999 standard actually uses null values in place of **all** despite confusion with regular null values

item-name	*color*	*number*
skirt	dark	8
skirt	pastel	35
skirt	white	10
skirt	**all**	53
dress	dark	20
dress	pastel	10
dress	white	5
dress	**all**	35
shirt	dark	14
shirt	pastel	7
shirt	white	28
shirt	**all**	49
pant	dark	20
pant	pastel	2
pant	white	5
pant	**all**	27
all	dark	62
all	pastel	54
all	white	48
all	**all**	164

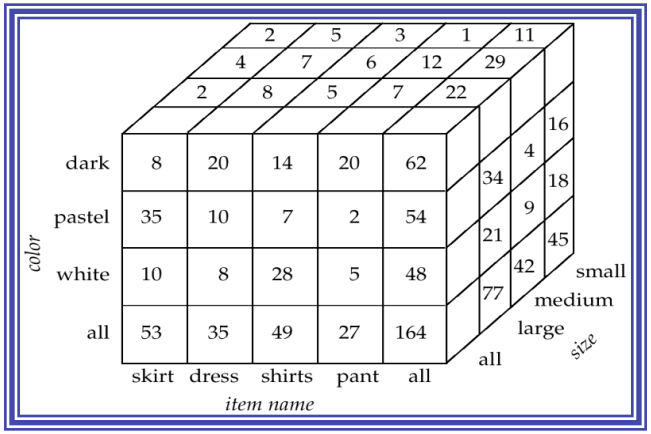
Data Cube

- A **data cube** is a multidimensional generalisation of a cross-tab
- Can have *n* dimensions; we show 3 below
- Cross-tabs can be used as views on a data cube



OLAP Queries

- **Pivoting**: changing the dimensions used in a cross-tab.
- **Slicing**: creating a cross-tab for fixed values only.
  - Sometimes called **dicing**, particularly when values for multiple dimensions are fixed.
- **Rollup**: moving from finer-granularity data to a coarser granularity.
  - the following is the rolling up on the attribute *size*



Rolling up on size

size: **all**

color

item-name	dark	pastel	white	Total
skirt	8	35	10	53
dress	20	10	5	35
shirt	14	7	28	49
pant	20	2	5	27
Total	62	54	48	164

- **Drill down**: The opposite operation - that of moving from coarser-granularity data to finer-granularity data

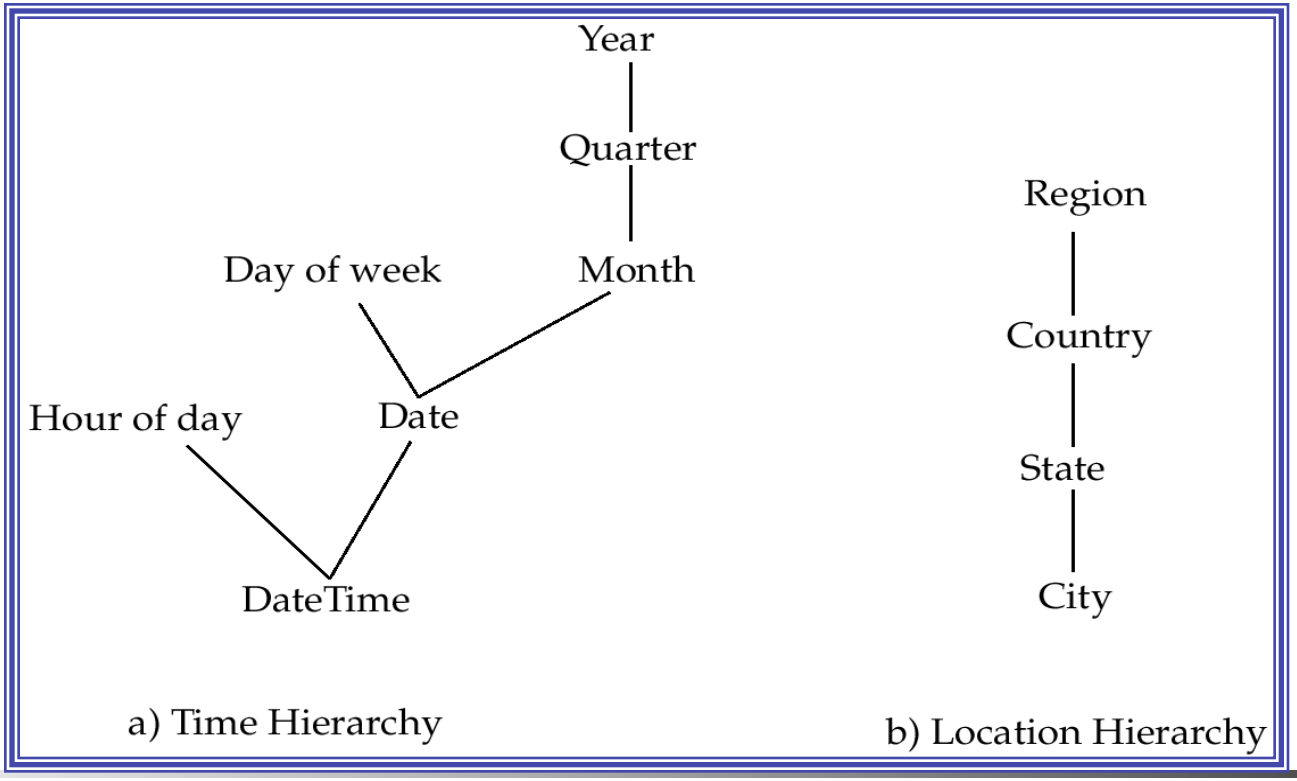
Slicing (Dicing) Query

- The general form of "slicing or dicing" query looks like

SELECT grouping attributes and aggregations
FROM fact table joined with zero or more dimension tables
WHERE certain attributes are constant
GROUP BY grouping attributes

Hierarchies on Dimensions

- **Hierarchy** on dimension attributes: lets dimensions to be viewed at different levels of detail
- E.g. the dimension DateTime can be used to aggregate by hour of day, date, day of week, month, quarter or year



Cross Tabulation With Hierarchy

- Cross-tabs can be easily extended to deal with hierarchies
- Can drill down or roll up on a hierarchy
- Size: ALL

category	*item-name*					
		dark	pastel	white	total	
womenswear	skirt	8	35	10	53	
	dress	20	10	5	35	
	subtotal	28	45	15		88
menswear	pants	14	7	28	49	
	shirt	20	2	5	27	
	subtotal	34	9	33		76
total		62	54	48		164

Data Mining

- Data mining is the exploration and analysis of large amount of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns in data.
  - Valid - The patterns hold in general.
  - Novel - We did not know the pattern beforehand.
  - Useful - We can devise actions from the patterns.
  - Understandable - We can interpret and comprehend the patterns.

What is a Data Mining Model?

- A data mining model is a description of a certain aspect of a dataset. It produces output values for an assigned set of inputs.
- Well studied examples:
  - Clustering
  - Regression
  - Classification
  - Frequent itemsets and association rules
  - Etc.

Classification

- Goal: learn a function that assigns a record to one of several pre-defined classes.
- Requirements on the model:
  - High accuracy
  - Understandable and interpretable by humans
  - Fast construction for very large training databases

Definitions

- Random variables $X_1, \ldots, X_k$ (*predictor variables*) and Y (*dependent variable*)
- X_i has domain $dom(X_i)$, Y has domain $dom(Y)$
- P is a probability distribution on $dom(X_1) \times \ldots \times dom(X_k) \times dom(Y)$; training database D is a random sample from P
- *A predictor* d is a function $d: dom(X_1) \ldots dom(X_k) \rightarrow dom(Y)$

Types of Variables

- *Numerical*: Domain is ordered and can be represented on the real line (e.g., age, income)
- *Nominal* or *categorical*: Domain is a finite set without any natural ordering (e.g., occupation, marital status, race)
- *Ordinal*: Domain is ordered, but absolute differences between values is unknown (e.g., preference scale, severity of an injury)

Classification Problem

- If Y is categorical, the problem is a *classification problem*, and we use C instead of Y. |dom(C)| = J, the number of classes.
- C is the *class label*, d is called a *classifier.*
- Let r be a record randomly drawn from P. Define the *misclassification rate* of d:

$$RT(d,P) = Prop(d(r.X_1, \ldots, r.X_k) != r.C)$$

- Problem definition: Given dataset D that is a random sample from probability distribution P, find classifier d such that RT(d,P) is minimised.

Regression Problem

- If Y is numerical, the problem is a *regression problem.*
- Y is called the dependent variable, d is called a *regression function.*
- Let r be a record randomly drawn from P. Define **mean squared error** rate of d:

$$RT(d,P) = E(r.Y - d(r.X_1, \ldots, r.X_k))^2$$

- Problem definition: Given dataset D that is a random sample from probability distribution P, find regression function d such that RT(d,P) is minimised.

Classification Example

- Training database:
  - Two predictor attributes: *Age* and *Car*-type (*Sport*, *Minivan* and *Truck*). Age is numerical, Car-type is categorical.
  - Class label indicates whether the person will buy the product
  - Dependent attribute is *categorical*

Age	Car	Class
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

Regression Example

- Example training database
  - Two predictor attributes: *Age* and *Car type* (**S**port, **M**inivan and **T**ruck)
  - Spent indicates how much the person spent for online shopping per month
  - Dependent attribute is *numerical*

Age	Car	Spent
20	M	$200
30	M	$150
25	T	$300
30	S	$220
40	S	$400
20	T	$80
30	M	$100
25	M	$125
40	M	$500
20	S	$420

Approaches for Classification

- Decision trees are one approach for classification.
- Other approaches include:
  - Linear Discriminant Analysis
  - Naïve Bayes
  - *k*-nearest neighbor
  - Logistic regression
  - Neural networks
  - Support Vector Machines
  - Adaboosting
  - Random forests
  - etc

Decision Trees

- A *decision tree* T encodes d (a classifier or regression function) in form of a tree
- A node t in T without children is called a *leaf node*. Otherwise t is called an *internal node* (except the root)
- Classification problem: Each leaf node is labeled with one class label c in dom(C)
- Each internal node has an associated splitting predicate. Common ones are binary predicates. Example predicate:
  - Age <= 20

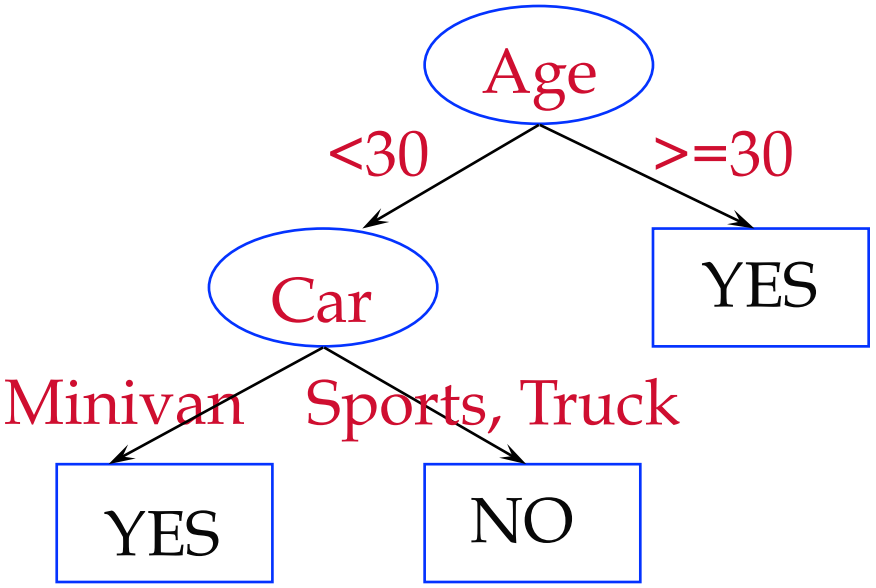
Example

Encoded classifier as rules:

If (age<30 and
carType=Minivan)
Then YES

If (age <30 and
(carType=Sports or
carType=Truck))
Then NO

If (age >= 30)
Then YES



Age	Car	Class
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

Construction of Decision Trees

- **Training set**: data samples in which the classification is already known.
- **Greedy top down** generation of decision trees.
  - Each internal node of the tree partitions the data into groups based on a **partitioning attribute**, and a **partitioning condition** for the node
  - **Leaf** node:
    - all (or most) of the items at the node belong to the same class, or
    - all attributes have been considered, and no further partitioning is possible.

Best Splits

- Pick best attributes and conditions to partition
- The purity of a set S of training instances can be measured quantitatively in several ways.
  - Notation: number of classes = k, number of instances = $|S|$, fraction of instances in class i: p_i (= (# of instances of class i in S)/$|S|$
- The purity(S) is defined as
  - Gini measure

$$\text{Gini}(S) = 1 - \sum_{i=1}^{k} p_i^2$$

  - or Entropy

$$\text{Entropy}(S) = -\sum_{i=1}^{k} p_i \log_2 p_i$$

  - When all instances are in a single class, both have values 0
  - Both reaches their maximum (for Gini measure is $1 - 1/k$) if each class has the same number of instances.

Best Splits cont'd

- When a set S is split into multiple sets S_i, i=1, 2, …, r, we can measure the purity of the resultant set of sets as (also known as average entropy or information):

$$\text{purity}(S_1, S_2, \ldots\ldots, S_r) = \sum_{i=1}^{r} \frac{|S_i|}{|S|} \text{purity}(S_i)$$

- The information gain due to particular split of S into S_i, i = 1, 2, …., r

$$\text{Information-gain}(S, \{S_1, S_2, \ldots., S_r) = \text{purity}(S) - \text{purity}(S_1, S_2, \ldots S_r)$$

Best Splits cont'd

- Measure of "cost" of a split (also known as intrinsic information):

$$\text{Information-content}(S, \{S_1, S_2, \ldots\ldots, S_r\})) = -\sum_{i=1}^{r} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- **Information-gain ratio** $= \dfrac{\text{Information-gain}(S, \{S_1, S_2, \ldots\ldots, S_r\})}{\text{Information-content}(S, \{S_1, S_2, \ldots\ldots, S_r\})}$

- The best split is the one that gives the maximum information gain ratio.

Finding Best Splits

- Categorical attributes (with no meaningful order):
  - Multi-way split, one child for each value
  - Binary split: try all possible breakup of values into two sets, and pick the best
- Continuous-valued attributes (can be sorted in a meaningful order)
  - Binary split:
    - Sort values, try each as a split point, e.g. if values are 1, 10, 15, 25, split at ≤1, ≤ 10, ≤ 15
    - Pick the value that gives best split
  - Multi-way split:
    - A series of binary splits on the same attribute has roughly equivalent effect

Decision-Tree Construction Algorithm

Procedure *GrowTree* (*S: Data set*, δ_p: *Real*, δ_s: *Nat*)
  Partition (*S*, δ_p , δ_s);

Procedure Partition (*S*)
  if (*purity* (*S*) < δ_p or |*S*| < δ_s) **then**
    return;
  for each attribute *A*
    evaluate splits on attribute *A*;
  Use best split found (across all attributes) to partition
    S into S_1, S_2, …., S_r,
  for *i* = 1, 2, ….., *r*
    Partition (S_i , δ_p , δ_s);

End of Lecture

- Summary
  - Data warehouse
  - OLAP: Online Analytical Processing
  - Introduction to Data mining: Classification and Decision Trees

- Reading
  - Textbook 6th edition, chapter 20
  - Textbook 7th edition, chapter 11
  - Tutorial on decision tree slides 1-28