Week 1 Intro to Data Mining

- Def: extraction of interesting patterns or knowledge from huge amounts of data.
- KDD: knowledge discovery in database

Process:

Data Cleaning ⇒ Data Integration ⇒ Data Selection ⇒ D. Trans.
⇒ Data. Mining ⇒ ~~Data~~ Pattern Evaluation ⇒ Knowledge Presentation

---

Week 2 Fandation Concepts for Data Mining

- Data Types
  - Attribute Types
    - Nominal / categorical
    - Binary
      - symmetric
      - asymmetric
    - Ordinal
    - Numeric Interval-Scaled (No true zero-point, °C)
    - Numeric Ratio-Scaled (With true zero-point, K)
  - Discrete vs. continuous
- measure functions
  - distributive, count(), sum(), min(), max() [can be partitioned]
  - algebraic, can be computed by a function with M arguments, each of which is obtained by applying a distributive ... AVERAGE() ...
  - holistic, ... median() rank(), mode().
- Basic statistical descriptions of a single data variable
  - Central Tendencies
    - arithmetic mean
    - weighted mean
    - median
    - mode ⇒ empirical for unimodal data at most moderately skewed
      mean − mode ≈ 3 (mean − median)
    - midrange = average (max, min)
    - skewness
  - Dispersion
    - range
    - quartiles
    - interquartile range (IQR) $= Q_3 - Q_1$
    - Five-# summary and boxplots
      - ↳ outliers $< Q_1 - 1.5IQR$ or $> Q_3 + 1.5IQR$
    - SD and Var
      Chebyshev's inequality:
      at least $(1 - \frac{1}{k^2}) \times 100\%$ of obs. $\leq k \cdot SD$ from the mean.

TWO Variables:

correlated ⇒ Pearson's correlation: $(-1, 1)$

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

MULTIVARIATE

Similarity & dissimilarity.

- dissimilarity matrix $d(i,j)=d(j,i)$, $d(i,i)=0$
  symmetric & triangular with zeros

$$sim(i,j)=1-d(i,j)$$

- Proximity for **NOMINAL** data:

  Method ① Simple matching $d(i,j)=\frac{p-m}{p}$ (p: vars, m: matched)

  ② map nominals to a large # of binary attributes

- Proximity for **BINARY** attributes

  - contingency table

	1	0	Sum
1	q	r	$q+r$
0	s	t	$s+t$
Sum	$q+s$	$r+t$	$q+r+s+t$

  - symmetric binary $d(i,j)=\frac{r+s}{q+r+s+t}$

  - asymmetric binary $d(i,j)=\frac{r+s}{q+r+s}$ (excluding both 0-0 pairs)

Jaccard coefficient / coherence:

$$sim(i,j)=1-d(i,j)=sim_{Jaccard}(i,j)=\frac{q}{q+r+s}$$

- --- of **NUMERIC** Data.

- min-max normalization, maps to $[0, 1]$

- metric { satisfies { identity of indiscernible $d(i,i)=0$; symmetry; triangle ineq.; non-negativity }

  e.g. Minkowski distance $d(i,j)=\sqrt[h]{|x_{i1}-x_{j1}|^h+\cdots+|x_{ip}-x_{jp}|^h}$

  Manhattan dist. $h=1$, L_1 norm

  Euclidean dist., $h=2$, L_2 norm

  Supreme distance, $h\to\infty$, L_∞/L_{max} norm

$$d(i,j)=\lim_{h\to\infty}\left(\sum_{f=1}^{p}|x_{if}-x_{jf}|^h\right)^{\frac{1}{h}}=\max_f^p|x_{if}-x_{jf}|$$

- **ORDINAL**: ① map ② ~~no~~normalize.
- **MIXED** types
  => bring all to a dissimilarity matrix.
  the dis of object i & j is defined as the average of the normalized dist. between each attribute

- cosine sim for sparse vectors

$$sim(x,y) = \cos(x,y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

where $\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$

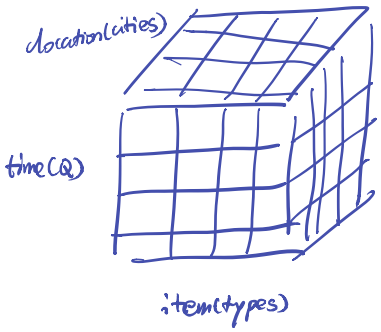
Week 3 Intro to Data Warehousing

- Data warehouse { subject-oriented
  integrated (cleaning, ETL (extract, transform, load))
  time variant
  nonvolatile

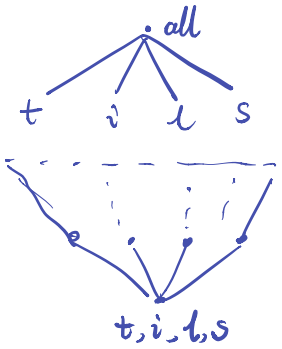
- Online Transaction Processing (OLTP) vs. Online Analytical Processing (OLAP)
  query day-to-day operations — data warehousing
  analytics to support decision making

DATA CUBE

- multidimensional
- axes = dimensions
- cube interior cells = measures



- n-dim => a base cuboid
- measures along 1 or 1+ dim => aggr to a cuboid with a subset of the original dims
- top most 0-D cuboid (apex), denoted all
- Lattice of cuboids forms a data cube.



0-D (apex)

1-D

2-D

3-D

4-D

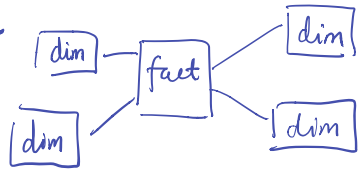
- n-dim, 2^n cuboids in the data cube lattice.
- how many level 2?

n = 10
r = 2
so $\frac{n!}{r!(n-r)!}$ ✓

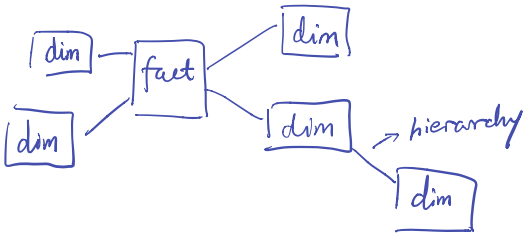
CUBE in a relational database

- dimension tables : item or time
- fact tables : containing measures & keys to each of the related dim table

- Star schema



- Snowflake schema



- fact constellations
  (multiple fact tables share dim tables,
  a collection of stars,
  galaxy schema or fact constellations)
- Concept hierarchy (low → high level)
- ROLAP (Relational OLAP), ~~star~~ schema database structure.
  /snowflake
- MOLAP (Multidimensional OLAP) column-oriented data storage architecture
  optimise rapid access to aggregate data & storage of sparse data.
- HOLAP (Hybrid OLAP) = ROLAP + MOLAP

- **OLAP operations**
  - rollup / drillup
    - by dimension reduction
      - aggregated,
      - ⇓ reduce dim
      - higher up to apex
    - by climbing up the concept hierarchy
      - aggregates the measures for that dim into bigger chunks.

    ⇕ opposite

  - drill down / roll down
    - by introducing additional dimensions
    - ~~st~~ by stepping down a concept hierarchy.

- Slice & Dice
  (cut out rectangular sections)
  - slice: cuts off one dimension of cube (not by aggregating but by selecting only one fixed value along 1 dim)
    - SQL: select
  - dice: cuts off a sub-cube. --- selecting multiple fixed values
- pivot (rotate)
- etc.

3 Kinds of Data warehouse applications (order of sophistication)

① Information processing ② analytical processing ③ data mining

- Efficient data cube computation
  - a lattice of cuboids
  - the bottom-most cuboid is the base cuboid
  - top-most apex contains only one cell.

  e.g. 3-D base cuboid: what is the ~~sum~~ Sum grouping by —, — & —.
  2-D --- sum grouping by — & —.
  0-D -- what is the total sum?

- Size of materialized data cube: 2^n cuboids
- when we also count cuboids generated for concept hierarchies for roll-ups

  L_i: # of levels in the concept i, then

  $$T = \prod_{i=1}^{n} (L_i + 1)$$

  say: item: $L_i = 1$
  time:

  L: level, hierarchy.

- Processing OLAP queries
  materialize whole / partially / none
- which queries to be selected to process the query?
  - contain at least the dimensions mentioned in the query.
  - Finer granularity data ~~could be~~ cannot be generated from coarser-granularity data.
- which would the best query?
  - Prefer a cuboid at the coarsest
  - Prefer a small cuboid or one with prebuilt efficient indexes

CUBE Materialization.

- Full materialization
- Cube Shell
  - compute all cuboids ~~with~~ $\leq k$ dim on ~~the~~ assumption ~~that~~ most investigations only need this shallow.
  - compute shell fragments by pre-computation & fill them in with run-time computation → semi-online strategy
- Iceberg cube : partial cube

ICEberg CUBE

- sparse cuboids, sparse cubes.
- low values in measures ⇒ little interest.
  iceberg cube ⇒ interest on top tip.
- The threshold is called the minimum support threshold or min_sup
- iceberg condition

e.g. 100-D base cuboid < 2 cells meeting the threshold condition Sum ≥ 10

A : $(a_1, a_2, \cdots, a_{100})$ with sum 10
B : $(a_1, a_2, b_3, \cdots, b_{100})$ with sum 10
$a_i \neq b_i$ for $i = 1, \cdots, 100$.

There are $2^{100}-1$ non-base cuboids so each cell in the base generates $2^{100}-1$ aggregate cells

$2 \times (2^{100}-1) - 4$
$= 2^{101} - 6$

$(a_1, a_2, *, *, \cdots, *) : 20$
$(a_1, *, *, \cdots, *) : 20$
$(*, a_2, *, \cdots, *) : 20$
$(*, *, *, \cdots, *) : 20$ → APEX

- closed cubes
A cell is closed cell if ∃ no other cell that is descendant (above it in the lattice) that has the same measure value as itself.

A closed cube materialize only closed cells.

e.g. a cell $(*, a_2, *, a_4, \cdots, a_{100})$ } ~~a cell~~
↓ ↓
aggregation over these 2 dims
that is a cell in 98-D cuboid of the datacube

---

4. Association Mining

- association mining
association rules
- data items are groups into transactions / itemsets.
  - find patterns of items that occur in a very high proportion of transactions

BASIC CONCEPTS

- itemset $I = \{I_1, \cdots, I_m\}$
- k-itemset $X = \{x_1, \cdots, x_k\}$ of cardinality k.
- T is a transaction, $T \neq \emptyset$, $T \in D$, D is a set of task-relevant transactions (or a dataset)
- ~~For~~ $\forall T$, ∃ a unique identifier TID
- $A, B \subseteq I$, T is said to contain A if $A \subseteq T$

- Support count: # of itemset A in dataset D is the cardinality of $\{T, \text{s.t. } A \subseteq T \,\&\, T \subseteq D\}$
- relative support or (just) Support: Support of A
- frequent: support $(A) \geq$ min_sup minimum support threshold

- association rule:
  $A \Rightarrow B$ where $A, B \subset I$, $A \neq \emptyset$ & $B \neq \emptyset$ and $A \cap B = \emptyset$.
  $A \Rightarrow B$ is said to hold in the dataset D with support s where s is the relative support of $A \cup B$ in D.
  $s = P(A \cup B)$
- confidence: $A \Rightarrow B$ with confidence c when $c = \dfrac{\text{Support}(A \cup B)}{\text{Support}(A)}$
  $c = P(B|A)$
- Strong: $A \Rightarrow B$ is strong when $A \cup B$ is frequent in D (satisfies min_sup) & the rule also satisfies min_conf (minimum confidence threshold.)

- strong rules are not always interesting.
- support & confidence measure association,

lift is used to measure correlation:

lift of rule $A \Rightarrow B$

$$\frac{\text{support}(A \cup B)}{\text{support}(A) \times \text{support}(B)}$$

lift = 1 => independent
lift > 1 => + correlated
< 1 => − correlated

- **CHI-Square**

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

$$e_{ij} = \frac{\text{count}(A = a_i) \times \text{count}(B = b_j)}{n}$$

expected frequency

observed frequency o_{ij}

$$\Rightarrow \chi^2 = \sum \frac{(\text{obs.} - \text{expec})^2}{\text{expected}}$$

degree of freedom $(r-1) \times (c-1)$

- $\chi^2 = 0$ iff expected & true # of obs. are all equal in all cells => all attrs indep.
- $\chi^2 \geq 0{,}05$, reject H_0, likely correlation.

$\chi^2 > 1$ => interesting! (thought don't know + or −)

- property : null-invariance

χ^2 & lift not null-invariance (0 ↑ change!)

confidence is null-invariance

Frequent Itemset Mining
1. Find all freq itemsets that occur at least min_sup × cardinality times
2. Use frequent itemsets to generate strong association rules that satisfy min_conf

APRIORI ALGORITHM

- iterative level-wise search for frequent itemsets where k-itemsets at level k are used to explore the k+1 itemsets at level k+1.

At each level, the transaction database is scanned to count items in transactions & to collect items that satisfy minimum support.

- apriori property

(closely related to iceberg condition)

All non-empty subsets of frequent item sets are also frequent.

⇕

If an item set is not frequent, then any of its supersets cannot be frequent.

a dataset of 3 transactions:

$T_1 = \{A, B, C, D\}$
$T_2 = \{A, B\}$
$T_3 = \{A, D\}$

with min support count $min\_sup = 2$

① Find item sets of size 1 that are frequent:

$L_1 = \{\{A\}, \{B\}, \{D\}\}$

② Call apriori_gen
pairwise join from L_1
check if the ~~check subset~~ freq.

$C_2 = \{\{A,B\}, \{B,D\}, \{A,D\}\}$

	$\{A,B\}$	$\{B,D\}$	$\{A,D\}$
T_1	✓	✗	✓
T_2	✓	✗	✗
T_3	✗	✗	✓
	2	1	2

$L_2 = \{\{A,B\}, \{A,D\}\}$
$C_2 = \{\{A,B,D\}\}$ not freq.

So $L = \{\{A\}, \{B\}, \{D\}, \{A,B\}, \{A,D\}\}$

• Efficient frequent itemset

- closed
  An itemset X is closed if no $Y \subseteq D$ s.t. $X \subset Y$ & Y has the same support count as $X \subset D$.
- closed frequent itemset.

- An itemset X is a maximal frequent itemset in D if X is frequent in D & there is no frequent Y in D such that $X \subset Y$.

D has 2 transactions

$\{\{a_1, \cdots, a_{100}\}, \{a_1, \cdots, a_{50}\}\}$

min_sup = 0.5, min support count 1

non-empty freq. itemsets in D are

$\{a_1\}, \cdots \{a_{100}\}, \{a_1, a_{100}\}, \cdots, \{a_1, a_2, a_3\}, \cdots \{a_1, \cdots, a_{100}\}$ all $2^{100-1} - 1$ of them

- closed itemsets of D are $\{a_1, \cdots, a_{100}\}$ with support count 1 & $\{a_1, \cdots, a_{50}\}$ with count 2, both frequent
- The only maximal frequent itemset in D is $\{a_1, \cdots, a_{100}\}$ with support count 1.

Extended data types

- Multi-dim patterns over nominal data

To distinguish the attributes for single-dimensional algorithms like apriori is to transform the values of nominal attributes to explicit attribute-value pairs & so to transfer all distinct nominal domains to the one nominal item domain for mining.

e.g. $\{a, b, c\} \Rightarrow \{A=a, B=b, C=c\}$

- Ordinal data
  - do not take account of any meaning order
  - so ordinal = nominal
    or when ordinal large = continuous

- Quantitative (continuous) data

① static => transfer quantitative attributes into discrete, predetermined nominal attributes.

② dynamic => { cluster attr into bins.
integrate values the freq. item-set discovery with a data warehouse cube.
or integrate a clustering method