Data types:

Numerical – numeric: real numbers

Categorical – nominal: unordered set

Categorical – ordinal: ordered set

Data set **types (4)**: **record data** (Relational records/Data matrix/Transaction data/Term-frequency vector (matrix) of text documents) **Graphs and Networks** (Transportation network/World Wide Web/Molecular Structures/Social or information networks)

**Ordered Data** (Video data/Temporal data: time series/Sequential Data: transaction sequences/Genetic sequence data) **Spatial, image and multimedia Data** (maps/image/video)

Structured Data properties: Dimensionality/Sparsity稀疏性/resolution解析度/distribution

Database rows → data objects; columns → attributes

Data set **attribute** 4 types: Nominal (hair color: black, gold)/ Binary/Ordinal/Numeric: Interval-scaled (no true 0 point. ○C/○F)/ Numeric: Ratio-scaled (have true 0 point. Count/length/monetary/ temp in ○K)

**Median(Approx):** 

Min approx. err=0 (when n/2=sum(freq), median=L1. Or when data in L2-L1 are unique & evenly distributed, eg: freq\_med=L2-L1)

Max approx. err=L2-L1 (when n/2-sum(freq)=freq\_med. Or when all data in that interval =L1)

**trimmed mean:** chop extreme value’s mean

**Mode:** 出现最多的value

**Unimodal(单峰):** mean – mode = 3\*(mean-median)

**positively skewed:** mean > median > mode

**negatively skewed:** mode > median > mean

**1sd**: 68% , 2sd: 95%, 3sd: 99.7%

**Var**(sample): 

**Var(**population): 

**Graphic Displays**: Boxplot (check dispersion, min Q1 median Q3 max) /Histogram/Quantile plot/Quantile-quantile (q-q) plot/Scatter plot(2-D)/Scatter plot matrix (n-D)

**Histo & Bar chart diffs:**

Histo: show distribution, x binned quantitative data, area denote value

Bar: compare variables, categorical data, can reorder, height denote value

**Data visualization**:

**Pixel-Oriented**: # of windows=# dim, color of pixel= values

**Geometric**: landscapes/Scatterplot Matrices/parallel coordinate

**Hierarchical:** **tree map**(hierarchical partitioning of screen into regions depending on attribute values)/**info cube**(semi transparent cube )/**cone-tree**(can’t avoid overlaps )/ **Dimensional Stacking**/ **Worlds-within-Worlds**(fix other param const, plot function wrt two most important param)

**Icon-based**: Chernoff Faces/ Stick Figures

**For complex data: Tag cloud** (importance: size/color, visualize world, phrase distribution)

**Social Networks** (visualizing non-numerical data)

**Data Similarity, Dissimilarity**

**Distance=1-similarity=dissimilarity**

**Proximity:** refer to either sim or dissim

**Similarity**:[0,1] 1=sim,0=no sim. Jaccard coef: q/(q+r+s)

**Distance(dissimilarity):**

[0,1] or [0,] 0=completely sim

**Numeric data:**

**Minkowski distance (L-p norm, 2D)**

****

p = 1 Manhattan distance (L1 norm) 差之和

p = 2 Euclidean distance (L2 norm) 欧几里德距离

p = inf supremum distance (Lmax norm) 距离为最大差值



**properties:**

d(i,j) > 00 if i ≠ j, and d(i, i) = 0 (Positivity)

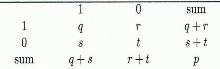
d(i,j) = d(j,i)

d(i,j) <= d(i,k) + d(k,j) (triangle inequality)

A distance that satisfies these properties is a **metric(度量)**

**set difference** (nonmetric dissimilarity)

**Binary data(contingency table):**

****

symmetric binary variable distance: (r+s)/(q+r+s+t)

asymmetric binary variable distance: (r+s)/(q+r+s)

**Jaccard coefficient**(asymmetric, **medical test**): q/(q+r+s) [0,1]

**Categorical data:**

Simple match:  (m # of matches, *p*: total # of variables)

Use a large number of binary attribute

**Ordinal data:**

Map rank to number [0,1] ((freshman, sophomore, junior, senior) to (0,1/3,2/3,1)), then use methods for interval-scaled variables (d(junior, senior) = 1/3)

**Two long vectors**(text, gene)**:**

Cosine **similarity**(null-invar): , dis=1-cos(sim)

**Mixed Type data**: weighted 

**KL Divergence** (difference between two probability distributions over the same variable x**, not distance measure**)

Not metric: asymmetric, not triangular inequality(*D*KL(*P*‖*Q*) does not equal *D*KL(*Q*‖*P*))), *DKL*(*P,Q*) *≥* 0 and *DKL*(*P* || *Q*) = 0 if and only if *P* = *Q*

Divergence of q(x) from p(x): (测用q(x)估计p(x)时的info lost.)

= the relative entropy of p with respect to q.=>





**Normalization**

Min-max normalization:

z-score normalization: 

Decimal scaling:

 *j* is the smallest integer s.t. max(|v’|) < 1

**Data problem:**

**Incomplete:** (missing data: ignore tuple/fill mean…)

**Noisy:** (outlier/noise/err: **binning** (partition into bins, smooth by mean median)/**regression/clustering** (detect and remove outlier)/**semi-supervised** (detect suspicious value, check by human

))

**Inconsistent:** 前后无法对应

**Intentional:** 伪装的missing data

**Data discrepancy detection(差异检测):**

Use **metadata元组** (domain, range, distribution)

Check field overloading

Check uniqueness/ consecutive/null rules

Use commercial tools (data scrubbing: simple knowledge to detect & clean, data auditing: analysis data, find rules& relations)

**Data integration(集成):**

Schema integration-data warehouse model (**star/snowflake/galaxy schema**): integrate metadata from diff sources

Entity identification: identify real world entity from diff sources

**Redundant data** (due to integration of multiple database)

2 types: object identification (same var diff name), derivable data (one var can get from others)

Solve: analyze cov/corr



**Correlation Analysis**:

**Chi-sq test** (corr for Categorical Data) [0,inf]:

H0: 2 dist indep, Chisq large→related

**Cov** (Two Variables): [-inf,inf], Sensitive with scale



=0 →indep

**Corr**: [–1, 1], not sensitive with scale

ρ12 > 0: A and B are positively correlated (X1’s values increase as X2’s)

Special eg: XY dep but corr=0: y=x^2, x[-2,2]

**Cov matrix**: semi-positive definite. Det(∑)>=0. Symmetric.





d dimensions

**Data Reduction(reduce data size) (6 types):**

**Parametric Data**: **Regression Analysis**(Linear and Multiple Regression/ Log-Linear Models, fit model, only keep estimated parameter, lossy): Estimate the probability of each point (tuple) in a multi-dim. space for a set of discretized attributes

**Non-Parametric Data**: **Histogram**(divide data into buckets, store average)/**Clustering**(cluster, store center and diameter, can hierarchical)/ **Sampling** (poor when data skewed, use small sample to represent population)/**Data Cube Aggregation**(basic cuboid: the lowest levelin cube)/**Data Compression (reduce both data size & dim)**

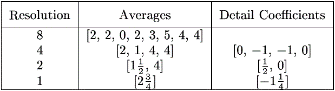
**Data Compression**

String compress: lossless

Audio (time sequence)/Video compress: lossy, can reconstructed small fragment of signal not whole.

Image compression (Wavelet Transform):

n-D signal, keep relative distance b/w obj at diff levels of resolution, let nature cluster more distinguishable.

****

**Data Transformation methods**:

Smoothing/ Attribute&feature construction/ Aggregation/Normalization/Discretization(hierarchy)/ Comcept hirerarchy generation(lower level concept → high level, numeric data: Discretization, nominal data: supervised & define order)

**Data Discretization Methods (5, partition→ smoothing):** **Correlation Binning:** top-down, unsupervised, equal width(distance)/equal depth (freg), smoothing by mean/bin boundaries

**Clustering** (classification): top-down, supervised, use entropy to split

**Correlation**: bottom-up, supervised, find best neighbor(chisq small) &merge

**Histogram**: top-down, unsupervised

**Decision-tree**: top-down, supervised

**Warehouse** is subject-oriented, integrated, time-variant, and nonvolatile collection of data

**A Multi-Tiered Architecture of warehouse**:

Top Tier: Front-End Tools Middle Tier: OLAP Server

Bottom Tier: Data Warehouse Server Data

**Dimensionality Reduction (4 types)**:

**Feature subset selection:** remove irrelative/redundant vars, use Heuristic search (best single attr under indep assump/best stepwise/optimal branch & bound)

**Feature extraction(transform from high dim → low):**

**PCA**(转坐标系: Aligned the canonical coordinate system with the nature axes of the data in Y, normalize all data&vector, lossless)

**Feature creatio**n =feature generation

**wavelet transformation**

**Data Cube Measures**

**Distributive**: apply function to aggregate values is the same with apply fun on all data without partition

**Algebraic**: not return one value, can be computed by an algebraic function on distributive measures, AVG = sum/count, min\_N, max\_N

**Holistic**: median, mode, rank

**OLAP indexing**

bitmap indexing: not good for high cardinality domains

join indexing: useful for star schema, foreign key and primary key

**OLAP Operations**

Roll up (drill-up): summarize data

Drill down (roll down): reverse of roll-up

Slice and dice: project and select

Pivot (rotate): reorient the cube, visualization, 3D to 2D planes

# cuboids in an n-dimensional cube with L levels



**Cube Technology**

A cube with 100 dimensions and contains only 2 base cells: {(a1, a2, a3, …., a100), (a1, a2, b3, …, b100)}

# aggregate cells if “having count >= 1”:

(a1, a2, a3 . . . , a100) will generate 2^{100} - 1 non-base cells

(a1, a2, b3, . . . , b100) will generate 2^{100} - 1 non-base cells

Among these, 4 cells are overlapped and thus minus 4 so we get: 2\*2^{100} - 2 - 4 =  2\*2^{100}-6

iceberg cells with condition: “having count >= 2”:

(a1, a2, \*, ..., \*):2(a1, \*, \*, ..., \*):2(\*, a2, \*, ..., \*):2(\*, \*, \*, ..., \*): 2

Iceberg cube → count >= min support

**Closed cell:** there is no descendant has the same measures; A cell c is closed if there exists no cell d, such that d is a descendant of c, and d has the same measure value as c{(a1, a2, \*, …, \*): 20, (a1, a2, a3 . . . , a100): 10, (a1, a2, b3, . . . , b100): 10}

**Closed cube**: only contain closed cell

**Multi-way array aggregation**

Bottom up, ABC → AB, can’t do Apriori pruning, no iceberg optimization, low-dim

Full cube or small number of dims

../Desktop/Screen%20Shot%202016-10-13%20at%206.15.49%20AM.png

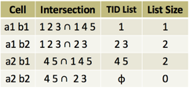
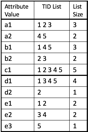
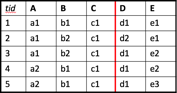
**BUC**

Top-down, root→A, sparse, iceberg pruning, if not satisfy min\_support, descendants pruned, high-dim



**High-D OLAP, Shell Fragment**

Tradeoff between the amount of pre-computation and the speed of online computation



Shell fragment don’t have to be disjoint

Given a database of T tuples, D dimensions, and F shell fragment size, the fragment cubes’ space requirement is:

**Frequent Pattern**

**Association rule**: x → y:(sup, conf)

Support: # x&y / # total

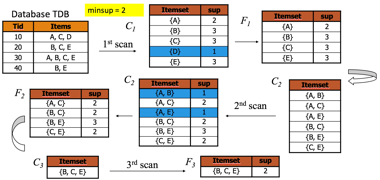
Confidence: # x&y / # x conditional probability

**Closed pattern**: no super-pattern Y>X with the same support as X, lossless

**Max-pattern**: frequent and no frequent super-pattern, lossy

**Apriori algorithm (k scan to get k-itemset)**

Apriori property:element cannot be negative



scan DB to get frequent 1-itemset

K = 1

F = {frequent pattern}

While Fk != None:

Ck+1 = candidate generated from Fk

Get Fk+1 by counting

k += 1

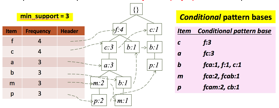
F = F | Fk

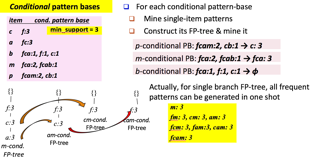
**Improve**

Reduce pass of database scan: partitioning, dynamic itemset counting

Shrink the number of candidates: DHP(hashing), pruning by support lower bounding, sampling

**FP-Tree**:





**FP Growth (2 scan)**

Scan DB once, find frequent 1-itemset, sort frequent items in descending order

Scan DB again, construct FP-tree

**Mine closed pattern**

item merging, if Y appears in every occurrence of x, merge y and x

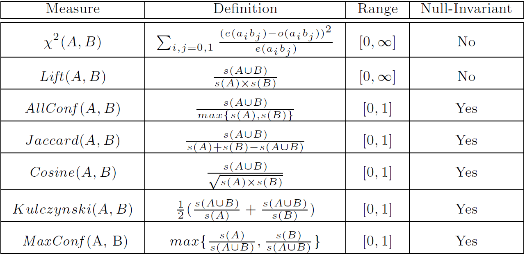
**Multi-Dimensional Associations**:

Single-dimensional rules - buys(X, “milk”) buys(X, “bread”)

Inter-dimension association rules - age(X, “18-25”) occupation(X, “student”) buys(X, “coke”)

Hybrid-dimension association rules - age(X, “18-25”) buys(X, “popcorn”) buys(X, “coke”)

**Pattern evaluation**



Kulczynski measure =(P(A|B) + P(B|A)) / 2 < epsilon (Negative Correlation)

**Negative pattern:** Negative Correlation, unlikely to happen together

If є = 0.01, we have (P(A|B) + P(B|A))/2 = (0.01 + 0.01)/2 < є

**Rare pattern:** very low sup, but interesting

**Pattern Anti-Monotonicity:** If an itemset S **violates** constraint c, so does any of its superset. Prun itemset.

***Data anti-monotone*:** In the mining process, if a data entry (transaction) *t* cannot satisfy a pattern *p* under *c*, *t* cannot satisfy *p*’s superset either. Prun data entry.

**Succinct:** if the constraint *c* can be enforced by directly manipulating the data (c可以通过直接操纵数据实施) (pattern w/o I, pattern contain I, min(s)<=v)

Sum & range not succinct, because cannot get sum at the beginning.

