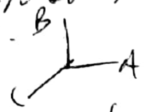
Data Cube Computation: 1) Sort, hash, & group tuples 2) Compute higher level aggregates from lower level & cache intermediate 3) Aggregate from child with smaller cardinality 4) Use Apriori to compute iceberg cubes. If a given cell does not meet min-sup neither do its descendents.

Multiway: Partition into chunks, cell address by "chunkID + offset", compute aggreg by visiting cells, optimize visit order so ~~partitions~~ partitions of agg cells in multiple cuboids can be computed simultaneously. 3-D array of ABC, 2-D & 1-D cuboids must be computed.



One BC chunk needs to be in memory at a time, for computation of all BC chunks. Simultaneously aggregate to each 2-D plane while one 3-D chunk is in memory. Minimum memory for holding all 2-D planes in chunk memory is whole AB plane $(40 \cdot 400)^{AB} + (40 \cdot 1000)^{AC \text{ column}} + (100 \cdot 1000)^{BC \text{ chunk}}$
When ~~AB~~ AB cuboid is being computed, A will have all its chunks in memory & B will have 1 chunk allocated at a time. Most effective when the product of cardinalities is moderate & data not too sparse. When dimensionality high or very sparse not good
BUC: Starts at apex & drills down. Can use apriori and share data partitioning costs. For each dimension, partition on d, to count # of tuples for each distinct val of d. Each distinct val forms its own partition. Iterate each partition, test for min-sup, if it passes recursively call BUC on d+1 next dim. On return from recursive call, continue on to next partition for d. Traverses tree. Computing AB does not help compute ABC. Process high cardinality dimensions first. Less skew is better
Star Cubing: Integrates top down & bottom up. Globally bottom up like Multiway with layers of top down to use Apriori with aggregate values at internal nodes. You can prune based on shared dimensions. Replace nodes not satisfying min-sup with *.

Multiway vs BUC: Multidim array vs. RDB, Full vs Iceberg, Simul Aggreg vs Part & sort
Can't ~~handle~~ handle big dim. Iceberg cannot be incrementally updated. Once pruned it is lost
Shell fragments: Compute fragments offline & combine them to dynamically answer online queries. Create inverted index for each fragment which is TID list for each attribute value. Find intersection of TID lists. Frag cube space required is
$O(T \lceil \frac{D}{F} \rceil (2^F - 1))$ F: size of fragments D of dims.
FP Tree: Find frequent 1-item sets. Sort freq items in order of descending support count. Scan each transaction, process items in L order, create branches, increment count. For conditional database find all branches containing that item. This is the conditional pattern base. Make conditional FP-tree column from frequent members of pattern base. Generate frequent patterns from conditional FP-tree, this column doesn't have to link consecutive items.
Apriori: k-itemsets are used to find (k+1) itemsets. First find frequent 1 item sets by scanning the database. If an itemset is not frequent then none of its supersets are. To get (k+1) join k itemsets need first (k-1) in common. Partition to find patterns. Any globally frequent pattern must be frequent in 1 partition
Vertical data format: Each item in it's own row with TID list
Apriori is breadth first, FP is depth first. ~~FP~~ FP says for frequent ~~~~ itemset P, restrict subsequent search to only items containing P. Mine conditional database recursively
Find frequent single items, ~~recursively grow FP~~ and partition DB based on each single item pattern, recursively grow freq patterns by doing above for each partitioned DB. Mining: recursive conditional until empty

support: $X \cup Y$ Confidence $X \to Y$ support/support(X) = $P(Y|X)$. A pattern is closed if it is frequent & there exists no super pattern with same support. A pattern is a max pattern if it is frequent & there is no frequent super pattern

$lift(B,C) = \frac{c(B \to C)}{s(C)} = \frac{s(B \cup C)}{s(B) \cdot s(C)}$ if it is 1 B & C are independent $X^2 = \sum \frac{O-E}{E}$

null transactions mess with lift & X^2. Jaccard & cosine are null invariant

Geometric techniques: Scatter, Parallel Icon: Chernoff, Stick figure Hierarchy: worlds, TreeMap

$X^2: [0, \infty)$ jaccard

	1	0
1	q	r
0	s	t

$sim(i,j) = \frac{q}{q+r+s}$ asymmetric disregard t

Dimensionality Reduction: Wavelet transform, PCA, attr subset, attribute construction. Numerosity: regression, log-linear, cluster, sampling

Nominal distance: $d(i,j) = \frac{p-m}{p}$ Difference in text: Cosine sim. minmax $= \frac{x - min}{max - min}(new_{max} - new_{min}) + new_{min}$

z-score $= \frac{x - \mu}{\sigma}$

$(a_1, a_2, \ldots, a_{10}): 1$, $(a_1, b_2, a_3, b_4, b_5, b_6, b_7, b_8, a_9, b_{10}): 1$ 2^{10} cuboids 3 closed cells: 2 base & $(a_1, *, a_3, *, * \ldots, a_9, *): 2$

A cell is closed if no descendant has same count. Aggregate cells? $2 \cdot 2^{10} - (2\ \text{base}) - 2^3$ duplicate

If min_sup = 2 # of aggregate cells? 8 3 dims in common

(gender, *) & (*, course) 400 courses, 2 gender. (F, c1) → (M, c1) need one space for course 2 for gender

$(a_1, a_2 \ldots, a_{10}): 1$ $(a_1, b_2, b_3, \ldots, a_9, b_{10}): 1$ $(a_1, c_2, c_3, \ldots, a_9, c_{10}): 1$ 2^3 cuboids contain 3 or less cells, 3 dims in common

only 1 cell for those 8, not 3 foreach base cell. Count once not 3 times. $2^3(3-1)$ dups

$3 \cdot 2^{10} - 3 - 2^3(3-1)$

for 3 base cells $3 \cdot 2^{10}$, then find dimensions common to all $3^{=y}$ and subtract $2 \cdot 2^y$. Then those common to any 2 say (j, k, l) subtract $(2^j - 1) - (2^k - 1) - (2^l - 1)$

Sum of largest x values not bounded, holistic. Sum of largest 50 values. In each partition maintain list of largest 50. Find largest 50 of those

sum(S.price) ≥ 45 is monotonic: as long as any subset of S has price ≥ 45 sum(S.price) ≥ 45

Sum(S.price) ≤ 45 is antimonotonic ~~as long as subset of S~~ if an item doesn't satisfy neither can any superset

avg(S.price) ≥ 30 can be converted to antimonotonic if items added in price descending order

conditional pattern base	conditional FP tree / conditional database	frequent patterns
all paths no suffix	frequent paths, can merge to get them no suffix	combinations include suffix

Manhattan $d(i,j) = |m - x_1| + |m - x_2| \ldots$ Euclidean $= \sqrt{|m_1 - f_1| + |m_2 - f_2|}$ Supremum is max

KL divergence measures diff between 2 prob distributions no vars using q to approx p $\sum_{x \in X} p(x) \ln \frac{p}{q}$

$\rho = \frac{Cov(A,B)}{\sigma_A \sigma_B}$

Need closed patterns to recover frequent patterns with counts