UIUC-CS412 "An Introduction to Data Warehousing and Data Mining" (Fall 2012)

Midterm Exam

(Wednesday, Oct. 24, 2012, 90 minutes, 100 marks, single sheet reference, brief answers)

Name: NetID: Score: **Answer Key**

1. [26] Data preprocessing.

   (a) [6] What are the value ranges of the following measures, respectively?

      i. z-score
      Answer: $(-\infty, +\infty)$

      ii. Pearson correlation co-efficient
      Answer: $[-1, 1]$

      iii. supremum distance (*i.e.*, L_∞ norm) for a set of m k-dimensional points.
      Answer: $(0, +\infty)$ will get full point. A better answer is: Let the suprenum distance between two points: (i, j) be $s(i, j)$, the range should be: $[min(s(i, j)), max(s(i, j))]$ for all (i, j) pairs of these m points.

   (b) [4] Name 4 visualization techniques that can visualize 5-dimensional data effectively.
   Answer: Any 4 popular visualization methods are OK, as long as they work for visualizing multidimensional data, such as stick figures, Chernoff faces, dimension stacking, parallel coordinates, scatter plot matrices (note: not scatter plot, which is 2-D only), etc.

   (c) [8] For each of the following similarity measures, give one good application example.

      i. Jaccard coefficient
      Answer: comparing patients based on their multiple medical testing results (principle: asymmetric variables, e.g., in medical tests, usually positive results are much infrequent than negative ones).

      ii. Cosine measure
      Answer: checking similarity between text documents (principle: bag of words forming long vectors for cosine measure to test).

      iii. covariance
      Answer: checking positive/negative correlations of numerical data.

      iv. χ^2 test
      Answer: checking positive/negative correlations of categorical data.

   (d) [8] Briefly distinguish the key difference between each of the following pairs of concepts?

      i. *tag cloud* vs. *worlds-within-worlds*
      Answer:
      tag cloud: visualizing text data, based on the relative word frequency;

worlds-within-worlds: visualizing multidimensional ($> 2D$) data, fix several dimension values, study how the measure distributes based on the values of the remaining 2-dimensions.

ii. *min-max normalization* vs. *z-score normalization*
Answer:
min-max: scale/normalize data to a specified (min, max) range;
z-score: scale/normalize data based on $z = \frac{x-\mu}{\sigma}$, where μ is mean and σ is standard deviation.

iii. *data reduction by parametric methods* vs. *data reduction by non-parametric methods*
Answer:
data reduction by parametric methods: reduce data by using parameters to describe the data, *e.g.*, regression model (results are parameters).
data reduction by non-parametric methods: reduce data by using reduced data sets to describe data, *e.g.*, feature selection, clustering, sampling.

iv. *Principal component analysis* vs. *feature selection*
Answer:
Principal component analysis: a dimensionality reduction method, the result is a set of orthogonal transformed features (*i.e.*, principal components).
feature selection: select some important (existing) features from the given feature set.

2. [24] Data Warehousing and OLAP for Data Mining

(a) [10] Suppose the base cuboid of a data cube contains 4 cells

$(a_1, a_2, c_3, \ldots, c_k)$, $(a_1, b_2, c_3, \ldots, c_k)$, $(b_1, a_2, c_3, \ldots, c_k)$, $(b_1, b_2, c_3, \ldots, c_k)$

where $a_i \neq b_i$ for any i

i. How many nonempty cuboids (including base cuboid) are there in this data cube?
Answer: 2^k since the cells contains k dimensions.

ii. How many (nonempty) aggregate closed cells are there in this data cube?
Answer: 5 (nonempty) aggregate closed cells. They are (not required to list them out but listing out will help them to answer questions correctly):
$(a_1, *, c_3, \ldots, c_k) : 2$,
$(b_1, *, c_3, \ldots, c_k) : 2$,
$(*, a_2, c_3, \ldots, c_k) : 2$,
$(*, b_2, c_3, \ldots, c_k) : 2$,
$(*, *, c_3, \ldots, c_k) : 4$

iii. How many (nonempty) aggregate cells are there in this data cube?
Answer: $4 \times 2^k - 4 - 4 \times 2^{k-2} - 3 \times 2^{k-2} = 9 \times 2^{k-2} - 4$.
Hint: "$(a_1, *, c_3, \ldots, c_k) : 2$" merges $2 \times 2^{k-2}$ cells into $1 \times 2^{k-2}$ cells and thus net loss is 2^{k-2} cells. There are 4 such cells. Thus their total loss is $4 \times 2^{k-2}$ cells.
"$(*, *, c_3, \ldots, c_k) : 4$" merges $4 \times 2^{k-2}$ cells into one 2^{k-2} cells and thus net loss is $3 \times 2^{k-2}$.

iv. If we set minimum support = 2, how many (nonempty) aggregate cells are there in the corresponding iceberg cube?

Answer: $5 \times 2^{k-2}$ since there are 5 aggregate closed cells, each will cover 2^{k-2} aggregate cells.

(b) [8] Suppose the BestBuy store would like to design a data cube to display the item sales in *count*, *average*, and in *boxplots* (containing *min*, *Q1*, *median, Q3*, and *max*) in multidimensional space (*e.g.*, by time and location). Explain which measures can be computed efficiently, but which ones cannot. Even for such difficult-to-compute measures, their approximations can often be computed efficiently. Outline how to efficiently compute one such an approximate measure.

Answer:

count, *average* (algebraic), *min*, and *max* are distributed or algebraic measures and they can be calculated efficiently.

Q1, *median*, and *Q3* are holistic measures and they cannot be computed efficiently.

Take median as an example, its approximation can be computed by interpolation using the formula: $l_i + \frac{N/2 - F_l}{F_i} \times (h_i - l_i)$ where l_i and h_i are the median interval (l_i, h_i) whose size is F_i, # of values lower than l_i is F_l, total count is N.

This is the same formula as what we described in the textbook:

$$median = L_1 + \left(\frac{N/2 - (\sum freq)_l}{freq_{median}} \right) width \tag{0.1}$$

where L_1 is the lower boundary of the median interval, N is the number of values in the entire data set, $(\sum freq)_l$ is the sum of the frequencies of all of the intervals that are lower than the median interval, $freq_{median}$ is the frequency of the median interval, and $width$ is the width of the median interval.

(c) [6] Bitmap index is often used for accessing a materialized data cube. If a cuboid has 6 dimensions, each has 50 distinct values, and it has in total 10000 cells. How many bit vectors should this cuboid have? How long each bit vector should be?

Answer: $6 \times 50 = 300$ bit vectors. Each bit-vector should be 10000 bits.

3. [22] Data cube implementation

(a) [8] Given the following four methods: *multiway array cubing* (Zhao, et al. SIGMOD'1997), *BUC* (bottom-up computation) (Beyer and Ramakrishnan, SIGMOD'2001), *StarCubing* (Xin et al., VLDB'2003), and *shell-fragment* approach (Li et al, VLDB'2004), list one method which is the best and another which is the worst (or not working) to implement one of the following:

(a) computing a dense full cube of low dimensionality (e.g., less than 6 dimensions),
Answer:
Best: multiway array cubing.
Worst: shell-fragment (since most part of cube is not precomputed)

(b) performing OLAP operations in a high-dimensional database (e.g., over 50 dimensions), and
Answer:
Best: shell-fragment
Worst: Any or the other three since they cannot compute high-D cubes.

(c) computing a large iceberg cube of around 10 dimensions.
Answer:
Best: star-cubing or BUC
Worst: multi-way array cubing

(b) [7] In a data cube, new datasets are often incrementally inserted into the base cuboid. One wants to incrementally compute a cube without recomputing the whole cube from scratch. (1) Can you do this for a *full cube*? and (2) can you do this for an *iceberg cube*? If you can, state how; but if you cannot, state why not.

Answer:

(1) Yes. Use aggregate function to update the measures of aggregated cuboids

(2) No. Since some previous cells below min-support were removed and the newly inserted data cannot compute correctly without computing the whole cube from scratch.

(c) [7] Sampling cube can be constructed by multi-dimensional aggregation over sampling data. Show the *confidence interval* for $\overline{x}$ (where x is a cell's sample set) is an algebraic measure. Discuss how to efficiently support drill-down given that some lower-level cells may be empty or contain too few data for reliable analysis.

Answer:

confidence interval is algebraic because it is computed based on the formula: $\overline{x} + t_c\overline{\sigma_x}$ where $\overline{\sigma_x} = s/\sqrt{l}$, where $\overline{x}$, s and l are distributed/algebraic measures.

Support query if the cell contains no sufficient data: use intra-cuboid expansion (using other cells in the cuboid) and inter-cuboid expansion (using rolling-up to get more values)

4. [25] Frequent pattern and association mining.

(a) [10] A data set shows 100 transactions in 5 days, each being summarized as a set of items associated with the number of transactions. Let $min\_sup = 0.5$ and $min\_conf = 0.7$.

date	items_bought	number of transactions
10/15	{p, a, b, c, m}	15
10/16	{b, e, f, p}	35
10/18	{p, a, c, k}	15
10/20	{a, b, e, p}	15
10/21	{p, a, g, e}	20

i. List the frequent 1-itemset (associated with their counts),
Answer:
$p : 100, e : 70, a : 65, b : 65$

ii. construct a frequent pattern (FP) tree for the dataset,
Answer: Any other trees will be ok if it is correct.

iii. Present all the frequent k-itemsets for the largest k:
Answer:
$peb : 50$

iv. Present two strong association rules (with support and confidence) containing the k items (for the largest k only):

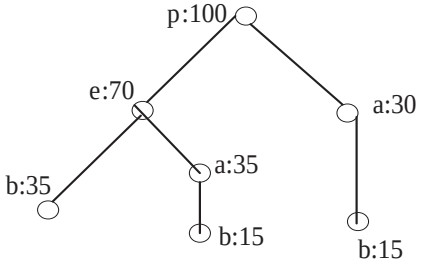

Figure 1: FP tree of a transaction DB

Answer:
$eb \rightarrow p(s: 50\%, c: 100\%)$
$bp \rightarrow e(s: 50\%, c: 77\%)$

(b) [8] Suppose in a transaction database, there are mc transactions containing both milk and coffee, $m\bar{c}$ transactions containing milk but not coffee, $\overline{m}c$ transactions containing coffee but not milk, and $\overline{mc}$ transactions containing neither. At what conditions, one can assert that milk and coffee are negatively correlated? At what conditions, one can assert that milk and coffee are positively correlated?

Answer:

The best is to use Kulczynski measure, if $kulc(m, c) > 0.5$ (or a predefined threshold δ), then m, c are positively correlated. if $kulc(m, c) < 0.5$ (or better, a predefined threshold ϵ), then m, c are negatively correlated.

Optionally one can add: Imbalance ratio can be used together with kulc to show if they are balanced or not.

Note: Using lift or χ^2 should get partial credit, such as 5/8, since they cannot handle null transaction cases well.

(c) [7] For mining frequent patterns, we should set different min-support thresholds for items in different groups (*e.g.*, bread vs. HDTV). Revise the Apriori algorithm to efficiently mine the set of frequent k-itemsets using group-based min-support.

Answer:

(1) partition items based on their min-support groups

(2) deriving group-based frequent 1-itemsets (*i.e.*, each group using its own min-support)

(3) when computing frequent k-itemsets, using the Apriori based on similar principles: a k-itemset can be frequent only if all of its corresponding frequent $(k-1)$-itemsets are all frequent

(4) for itemsets containing only the same grouped items, use the this group's min-support threshold. For itemsets containing items from different groups, use the smallest one among all the participating group's min-support thresholds. Reason: The larger one will fill-out valuable patterns (*e.g.*, HDTV will be filled out) and the small one will not encourage more combinations of bread (say B1, B2) to show up since they are controlled by their corresponding larger min-support in their corresponding groups.