CS412 "An Introduction to Data Warehousing and Data Mining" (Fall 2007)
Midterm Exam

(Monday, Oct. 15, 2007, 90 minutes, 100 marks, single sheet reference, brief answers)

Name: NetID: Score:

1. [30] Data preprocessing.

   (a) [6] For data visualization, there are three classes of techniques: (i) geometric techniques, (ii) hierarchical techniques, and (iii) icon-based techniques. Give names of two methods for each of these techniques.

   Answer:

   (i) geometric techniques: scatterplot matrices, parallel coordinates, landscapes

   (ii) hierarchical techniques: dimension stacking, tree maps, cone trees, info cube

   (iii) icon-based techniques: Chernoff face, stick figures, color icons, tile bars

   (b) [4] What are the value ranges of the following correlation measures, respectively?

   i. χ^2:
   Answer: $[0, +\infty)$

   ii. *Pearson correlation coefficient*:
   Answer: $[-1, +1]$

   (c) [8] Name four methods that perform effective *dimensionality reduction* and four methods that perform effective *numerosity reduction*.

   Answer:

   (i) Dimensionality reduction: SVD, PCA, decision tree, feature subset selection, feature creation, one may also count: wavelet/Fourier transformation

   (ii) Numerosity reduction: data compression, regression, clustering, sampling, binning, discretization, histogram, data cube aggregation, and also wavelet/Fourier transformation

   (d) [8] For the following group of data

   $$100, 200, 400, 800, 1500$$

   i. Calculate its mean and variance.
   Answer:
   (i) mean: $\mu = (100 + 200 + 400 + 800 + 1500)/5 = 600$
   (ii) variance : $\sigma^2 = 1/5[(100 - 600)^2 + (200 - 600)^2 + (400 - 600)^2 + (800 - 600)^2 + (1500 - 600)^2] = 260000$

   ii. Normalize the above group of data by min-max normalization with min = 0 and max = 10; and
   Answer:
   100: 0
   200: $((200 - 100)/(1500 - 100)) \times 10 + 0 = 0.714$
   400: $((400 - 100)/(1500 - 100)) \times 10 + 0 = 2.143$
   800: $((800 - 100)/(1500 - 100)) \times 10 + 0 = 5$
   1500: 10

iii. In z-score normalization, what value should the first number 100 be transformed to?
Answer:
100: $(100 - \mu)/\sigma = (100 - 600)/\sqrt{260000} = -0.98$

(e) [4] What are the best distance measure for each of the following applications:

(i) driving distance between two locations in Downtown Chicago,

Answer: Manhattan distance

(ii) compare similar diseases with a set of medical tests,

Answer: Dissimilarity for asymmetric binary variables or Jaccard coefficient, i.e., $(b+c)/(a+b+c)$

(iii) find similar web documents

Answer: Cosine measure of two vectors, i.e., the inner product of two feature vectors, each representing the features (such as keywords or terms) of a document.

2. [13] Data Warehousing and OLAP for Data Mining

(a) [3] Suppose a cube has 10 dimensions and the i-th dimension has M_i levels (not including *all*), *how many cuboids* does this cube contain (including base and apex cuboids)?

Answer: $\Pi_{i=1}^{10}(M_i + 1)$

(b) [4] Give two examples for each of the following two kinds of measures: (i) algebraic, and (ii) holistic.

Answer:

(i) algebraic: average, variance

(ii) holistic: median, Q_1, rank

(c) [6] Suppose the academic office of UIUC wants to build a Student_Record data warehouse with the following information: *student, major, course, department, grade*, and would like to calculate student GPA, major_gpa, etc.

(i) Draw a **snowflake schema** diagram (sketch it, and make your implicit assumptions on the levels of a dimension and the necessary measures).

Answer:

There could be many different answers in the design. One possible answer could be as follows.

Dimensions:

- Student(name, major, birth_place (...), ...),
- Department (dname, college, head, ...),
- Course(cname, cno, credit, instructor, ...),
- Time (semester, year, ...)

Measure: total_# = count(*), GPA = avg(grade), ...

The dimension tables should be linked to the fact table.

(ii) If one would like to start at the Apex cuboid and find top_10 students in each department in the College of Engineering based on their GPA up to Spring 2007, what are the **specific OLAP operations** (e.g., roll-up on which dimension from which level to which level) that one should perform based on your design?

Answer: OLAP operations:

Drill down on Department from * to College-level

Drill down on Time dimension from * to year-level

Dice on (i.e., select) college = "Engineering" and Year = "2007"
Drill down on Time to season and slice on season = "Spring"
Drill down on Department to the department-level
Drill down on Student dimension to student name (or ID)
Select top_10 GPA values, and print the corresponding student names

3. [25] Data cube implementation

(a) [10] Assume a base cuboid of N dimensions contains only p (where $p > 3$) nonempty base cells, and there is no level associated with any dimension.

i. What is the *maximum number of nonempty cells* (including the cell in the base cuboid) possible in such a materialized datacube?
Answer:
Each cell generates 2^N cells. So p cells will generate in total $p \times 2^N$ cells. However, the p cells at the Apex cuboid are merged into one, i.e., we need to minus $p-1$ cell count. Thus the maximum number of cells generated will be:

$$p \times 2^N - p + 1$$

ii. If the minimum support (i.e., iceberg condition) is 3, what is the *minimum number of nonempty cube cells* possible in the materialized iceberg cube?
Answer:
Each cell generates 2^N cells. However, these cells may not have to be merged together at the k dimensional cuboids (for $1 < k < N$) to increase its count. Until at the last moment, i.e., in the Apex cuboid, they have to be merged into one. In this case, it will generate one cell with count of $p > 3$. Thus the minimum number of cells generated will be: 1.

iii. If the minimum support is 2, what is the *maximum number of nonempty cells* possible in the materialized iceberg cube?
Answer:
Note: The following answer only works for 2-Dimension. Cases with higher dimension are very tricky.
The earliest stage that these cells may be merged together to form support 2 cells is at the $N-1$ dimensional cuboids. At this time, the maximum number of cells that one may form at each plane is $\lfloor \frac{p}{2} \rfloor$. Thus we can view the original cells are essentially $\lfloor \frac{p}{2} \rfloor$ support two cells at the total of N such $(N-1)$-dimensional spaces. Similar to question 1, they may generate in total $N \times \lfloor \frac{p}{2} \rfloor \times 2^{N-1}$ cells, except the Apex cuboid has p cells merged into one. Thus the maximum number of cells generated will be:

$$N \times \lfloor \frac{p}{2} \rfloor \times 2^{N-1} - p + 1$$

(b) [10] 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: Array-cubing. Worst: Shell-Fragment

(b) computing a large iceberg cube of around 8 dimensions, and
Answer: Best: BUC or StarCubing. Worst: Array-Cubing

(c) performing OLAP operations in a high-dimensional database (e.g., over 50 dimensions).
Answer:
Best: Shell-Fragment
Not-working: all the other three: BUC, StarCubing, and Array-Cubing
Note: Answering any one of the three will be OK

(c) [5] Suppose a disk-based large relation contains 100 attributes. What is the minimum number of database scans in order to derive a generalized relation by *attribute-oriented induction*?

Answer:

Two scans: one preparing for generalization, one performing generalization (which will derive a small, memory-resident prime-relation)

or

$1 + \delta$ scan since the first scan can be replaced by a δ scan or sampling.

4. [30] Frequent pattern and association mining.

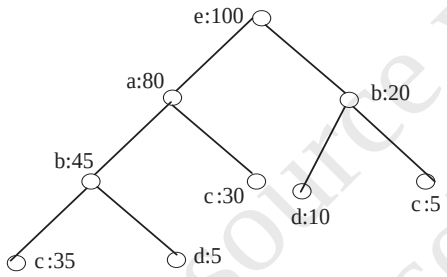

Figure 1: FP tree of a transaction DB

(a) [8] A database with 100 transactions has its FP-tree shown in Fig. 1. Let $min\_sup = 0.5$ and $min\_conf = 0.8$. Show

i. c's conditional (i.e., projected) database:
Answer: bae: 35, ae: 30, be: 5

ii. all the frequent k-itemsets for the largest k:
Answer: k = 3; cae: 65

iii. two strong association rules (with support and confidence) containing the k items (for the largest k only):
Answer: Note: ac: 65, ae: 80, ce: 70, a: 80, c: 70, e: 100. Thus we have
$c \rightarrow ae \;\; s = .65, \sigma = 65/70 = .92$
$a \rightarrow ce \;\; s = .65, \sigma = 65/80 = .81$
$ae \rightarrow c \;\; s = .65, \sigma = 65/80 = .81$
$ac \rightarrow e \;\; s = .65, \sigma = 65/65 = 1.00$
$ce \rightarrow a \;\; s = .65, \sigma = 65/70 = .92$

(b) [8] To further improve the Apriori algorithm, several *candidate generation-and-test* methods are proposed that *reduce the number of database scans* at mining. Briefly outline two such methods.

Answer: Any of the following algorithms will count:

1. Partitioning: partition DB into k portions, each fit in memory; mine each local partition; merge freq-itemsets; then one more scan DB to consolidate the global patterns.

2. Hashing: First scan, count freq-1 and hash 2-itemsets into buckets. If the bucket count < threshold, all the 2-itemsets in it are infreq.

3. DIC: Scan to count freq-1, and if 1-freq., start counting cand-2-itemsets, and so on.

(c) [6] Suppose a transaction database contains N transactions, ct transactions contain both coffee and tea, $c\bar{t}$ transactions contain coffee but not tea, $\bar{c}t$ transactions contain tea but not coffee, and $\bar{c}\bar{t}$ transactions contain neither tea nor coffee.

(i) What is the *null-invariance* property?

Answer: A measure not influenced by the count of $\bar{c}\bar{t}$ (i.e., those containing neither coffer nor tea).

(ii) give the names or definitions of three *null invariant measures*.

Answer: all-conf, coherence, Kulczynski, cosine, max-conf

(d) [8] Suppose a manager is interested in only the *frequent patterns* (i.e., *itemsets*) that satisfy certain constraints. For the following cases, state the characteristics of *every constraint* in each case and how to mine such patterns efficiently.

i. The price difference between the most expensive item and the cheapest one in each pattern must be within \$100.
Answer:
$C : range(S.price) \leq \$100$ is antimontonic.
Method: Push C into iterative mining, toss S if it cannot satisfy C.

ii. The sum of the profit of all the items in each pattern is between \$10 and \$20, and each such item is priced over \$10.
Answer:
$C_1 : min(S.price) > \$10$ is succinct, or data anti-monotone.
Method: Push C_1 into iterative mining, select only items satisfying C_1
$C_2 : sum(S.profit) \geq \$10$ is ***convertible*** monotone, $C_3 : sum(S.profit) \leq \$20$ is ***convertible*** antimonotone, if items within a transaction is sorted in profit ascending order.
Method: Push C_3 into iterative mining, toss S if it does not satisfy C_3 and check C_2, and once its satisfies, no more checking needed.
[Note: Both C_2 and C_3 can be used as data antimonotone constraints to prune t_i if the remaining items in t_i with the current S cannot satisfy C_2.]

5. [3] (Opinion).

(a) I ☐ like ☐ dislike the exams in this style.
Answer: L: 52, D: 31, not sure: 1

(b) In general, the exam questions are ☐ too hard ☐ too easy ☐ just right.
Answer: H: 52, E: 0, R: 32

(c) I ☐ have plenty of time ☐ have just enough time ☐ do not have enough time to finish the exam questions.
Answer: P: 10, E: 36, N: 38