UIUC-CS412 “An Introduction to Data Warehousing and Data Mining” (Fall 2016)

Midterm Exam

(Thursday, Oct. 13, 2016, 90 minutes, 100 marks, single sheet reference, brief answers)

1. [31] Data preprocessing.

   (a) [5] Names five data visualization methods that can visualize 4-dimensional data effectively.

   Answer: Most of the visualization methods, such as stick figure, Chernoff face, dimension stacking, parallel coordinates, scatter plot matrices (note: not scatter plot, which is 2-D only), etc.

   (b) [9] What is the best distance measure for each of the following applications:

   i. [2] Find the maximum difference between any numerical attribute of two vectors

   Answer: Supremum (i,e., L_∞-norm)

   ii. [2] Find whether two text documents are similar

   Answer: Cosine

   iii. [2] Measure the difference between two probability distributions over the same random variable x

   Answer: KL-divergence

   iv. [3] Find a sequence of words that occur much more frequently than *expected* in a large corpus

   Answer: χ^2-test

   (c) [9] What are the value ranges of the following measures, respectively?

   i. [2] χ^2 test statistics

   Answer: $[0, +\infty)$

   ii. [2] Pearson correlation coefficient

Answer: $[-1, +1]$

iii. [2] Kullback-Leibler (KL) divergence

Answer: $[0, +\infty)$

iv. [3] supremum distance (*i.e.*, L_∞ norm) for a *given set* of m points of k numerical dimensions

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.

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

i. [2] *scatter plot* vs. *boxplot*

Answer:
scatter plot: each pair of values is a pair of coordinates and plotted as points in the plane
boxplot: uses min, Q1, median, Q3, max (or whiskers, and outliers) to describe data distribution/dispersion

ii. [2] *covariance* vs. *Pearson correlation coefficient*

Answer:
Pearson correlation coefficient (X, Y) = covariance (X, Y) $/\sigma_X \sigma_Y$
(σ: standard deviation)

iii. [2] *Principal Component Analysis (PCA)* vs. *feature selection*

Answer:
Principal component analysis: a dimension 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.

iv. [2] *wavelet transform* vs. *lossless data compression*

Answer:
Wavelet transform splits data into average (smooth) and difference, transformation preserves shape. When it is used to compress data, only a fraction of strongest of the wavelet coefficients will be stored. Therefore, it is lossy data compression.

2. [27] Data Warehousing and OLAP for Data Mining

(a) [12] Suppose the base cuboid of a data cube contains two cells

$(a_1, a_2, a_3, a_4, \ldots, a_{10}) : 1$, $(a_1, b_2, a_3, b_4, \ldots, b_{10}) : 1$

where $a_i \neq b_i$ for any i

i. [3] How many nonempty cuboids are there in this data cube?

Answer: $\mathbf{2^{10}}$. Since we have 10 dimensions with no concept hierarchy, there are 2^{10} cuboids and all of them should not be empty.

ii. [3] How many (nonempty) aggregate closed cells are there in this data cube?

Answer: 1. There are 3 closed cells, including the two base cells and $(a_1, *, a_3, *, a_5, *, a_7, *, a_9, *)$. But only the latter one is a **aggregated** closed cell.

iii. [3] How many (nonempty) aggregate cells are there in this data cube?

Answer: 2014. For each base cell, there are $2^{10}-1$ aggregated cells. However, there are 2^5 cells that are counted twice since there are 5 common dimensions. Therefore, the total number of nonempty aggregate cells is $2 \cdot (2^{10} - 1) - 2^5 = 2014$.

iv. [3] If we set minimum support = 2, how many (nonempty) aggregate cells are there in the corresponding iceberg cube?

Answer: $\mathbf{2^5}$. These two base cells have common value in 5 dimensions; therefore, there are 2^5 nonempty cells with support = 2 and all of them are aggregate cells.

(b) [9] Which of the following measures are algebraic measures? Use one sentence to explain each of your selection and non-selection.

(i) [3] *standard deviation*;

Answer: Yes. Standard deviation can be calculated by *count()*, *sum()*, *sum(x^2)*. All of them are distributed measures.

(ii) [3] average of Q1 and Q3 (Note: Q1: the first quantile); and

Answer: No. Like *median()*, *Q1()* and *Q3()* are both holistic measures, which requires sorting all the data.

(iii) [3] sum of bottom-5.

Answer: Yes. In each partition, you can maintain a list of *bottom-5* data. Applying *bottom-5()* on these lists will give you the *bottom-5* of the whole data.

(c) [6] Bitmap index is often used for accessing a materialized data cube. If a cuboid has 8 dimensions, each has 50 distinct values, and it has in total 50000 cells.

(i) [3] How many bit vectors should this cuboid have?

Answer: 400. You need a bit vector for each distinct value in each dimension.

(ii) [3] How long each bit vector should be?

Answer: 50000. The length of a bit vector should be the total number of cells since you need an entry for each of them.

3. [17] Data cube implementation

(a) [8] Which of the following algorithms: (i) BUC, (ii) Multiway array cubing, and (iii) Shell-Fragment, cannot support the following operations efficiently? and explain why.

i. [4] Computing an iceberg cube

Answer: Multiway Array cubing cannot support iceberg computation since it computes aggregation bottom-up and Apriori principle/pruning cannot be applied here. (Note that shell-fragment can support)

ii. [4] Supporting efficient OLAP query processing on a large dataset with 50 dimensions (*i.e.*, attributes)

Answer: Multiway Array and BUC – 50 dimensions is too many for both algorithms to support.

(b) [9] Suppose a data relation has 100 attributes and 10^6 tuples. Each attribute has 50 distinct values. Suppose each cell takes 16 bytes of space, and the shell-fragments are all 4 dimensions.

(i) [5] What is the size (in bytes) of one pre-computed shell-fragment of size 4?

Answer: $\mathbf{16 \cdot (1 + 4 \times 50 + 6 \times 50^2 + 4 \times 50^3)}$ **Bytes.**
For the 0-D (apex) cuboid, there is 1 cell.
For each 1-D cuboid, there are $50 * 1 * 1 * 1 = 50$ cells. There are 4 such cuboids.
For each 2-D cuboid, there are $50 * 50 * 1 * 1 = 50^2$ cells. There are 6 such cuboids.
For each 3-D cuboid, there are $50 * 50 * 50 * 1 = 50^3$ cells. There are 4 such cuboids.
The total size of these cells are $16 \cdot (1 + 4 \times 50 + 6 \times 50^2 + 4 \times 50^3)$, or equivalently, $16 \cdot ((50 + 1)^4 - 50^4)$.
Note that if we also count the base cells, since the number of tuples are smaller than 50^4, we can actually store the data in 10^6 cells. We will add $16 \cdot 10^6$ to the size.

(ii) [4] If an OLAP query contains 2 instantiated variables and 6 inquired variables, what is the number of shell fragments this query has to access in the best case and in the worst case, respectively?

Answer: Best case: 2, worst case: 8. There are 8 dimensions we care about. In the best case, these 8 dimensions happen to be in 2 shell fragments since there are 4 dimensions each. In the worst case, all these 8 dimensions are in different shell fragments and we need to access 8 shell fragments.

4. [25] Frequent pattern and association mining.

(a) [6] Suppose a store has mined frequent patterns for 2016 up to September. Briefly describe one efficient **incremental** mining method which derives patterns that cover all the transaction data (including the current month), without re-mining the entire transaction database.

Answer: The key is not to re-mine the whole database. Patterns that do not appear in either set (old DB up to 2016.9 and new DB: 2016.10 to now) cannot be globally frequent. So, we only need to (1) mine FPs from the new DB; (2) get counts of those patterns frequent in new DB but not in old DB by scanning the old DB once, (3) get counts of those that are frequent in old DB by not in new DB. The merge counts will be used to just which patterns are globally frequent.

(b) [8] A database has 5 transactions. Let $min\_sup = 0.6$ and $min\_conf = 0.7$.

trans_id	items_bought
100	{K, A, D, B, C}
200	{D, A, E, F}
300	{C, D, B, E }
400	{B, A, C, K, D}
500	{B, G, C}

i. [2] List the frequent k-itemset for the largest k, and

Answer: $k = 3$
Frequent $3-$itemset: $\{B, C, D\}$ with $support = 3/5 = 0.6 \geq min\_sup$.

ii. [6] all the strong association rules (with support and confidence) for the following shape of rules:
$\forall x \in transaction,\ buys(x, item_1) \wedge buys(x, item_2) \ \Rightarrow\ buys(x, item_3). \qquad [s, c]$

Answer:
$\forall x \in transaction,\ buys(x, B) \wedge buys(x, C) \ \Rightarrow\ buys(x, D). \qquad [0.6, 0.75]$
$\forall x \in transaction,\ buys(x, B) \wedge buys(x, D) \ \Rightarrow\ buys(x, C). \qquad [0.6, 1]$
$\forall x \in transaction,\ buys(x, C) \wedge buys(x, D) \ \Rightarrow\ buys(x, B). \qquad [0.6, 1]$

(c) [5] The definitions of two measures, *lift* and *cosine*, look rather similar as shown below,

$$\mathit{lift}(A,B) = \frac{s(A \cup B)}{s(A) \times s(B)}$$

$$\mathit{cosine}(A,B) = \frac{s(A \cup B)}{\sqrt{s(A) \times s(B)}}$$

where $s(A)$ is the *relative* support of itemset A. Explain why one of these two measures is *null-invariant* but the other is not.

Answer: A measure is null-invariant if the value of the measure does not change with the number of null-transactions.
cosine is null-invariant while *lift* is not.
Let n be the total number of transactions, and $\mathit{count}(\neg(A \cup B))$ be the number of null-transactions.

$$\mathit{lift}(A,B) = \frac{s(A \cup B)}{s(A) \times s(B)} = \frac{\frac{\mathit{count}(A \cup B)}{n}}{\frac{\mathit{count}(A)}{n} \times \frac{\mathit{count}(B)}{n}} = \frac{\mathit{count}(A \cup B) \times n}{\mathit{count}(A) \times \mathit{count}(B)} = \frac{\mathit{count}(A \cup B) \times (\mathit{count}(A \cup B) + \mathit{count}(\neg(A \cup B))}{\mathit{count}(A) \times \mathit{count}(B)}$$

$$\mathit{cosine}(A,B) = \frac{s(A \cup B)}{\sqrt{s(A) \times s(B)}} = \frac{\frac{\mathit{count}(A \cup B)}{n}}{\sqrt{\frac{\mathit{count}(A)}{n} \times \frac{\mathit{count}(B)}{n}}} = \frac{\mathit{count}(A \cup B)}{\mathit{count}(A) \times \mathit{count}(B)}$$

We can clearly see that *cosine* is invariant with the number of null-transactions, while *lift* is not.

(d) [6] Suppose a WalMart manager is interested in only the *frequent patterns* (i.e., *itemsets*) that satisfy certain constraints. For the following cases, state the characteristics (*i.e.*, categories) of *every constraint* in each case and how to mine such patterns most efficiently.

i. [3] The average price of all the items in each pattern is greater than \$40.

Answer: C: $\mathit{avg}(S.\mathit{price}) > 40$.
This constraint is **strongly convertible**.

- It can be converted to **anti-monotonic** if the items are sorted in **descending** order of their prices.
  Method: Push C into iterative mining, toss S if it cannot satisfy C because adding any remaining items into S will not satisfy C anymore.
- It can be converted to **monotonic** if the items are sorted in **ascending** order of their prices.
  Method: Push C into iterative mining, if S satisfies C, no more checking is needed because adding any remaining items into S will satisfy C.

ii. [3] The sum of the price of all the items with profit over \$5 in each pattern is at least \$50.

Answer: C_1: $min(S.profit) > 5$ is **succinct**, or **data anti-monotone**.
Method: Push C_1 into iterative mining, select only items satisfying C_1.
C_2: $sum(S.price) \geq 50$ is **monotone**.
Method: Push C_2 into iterative mining, stop checking once S satisfies C_2.