CS 5525 Assignment 4

Due 19th Nov 2020

Problem 1. Let us consider the Boosting algorithm in which all the data points are initially given uniform weights. Given below is the figure. **[15 points]**

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Y	1	1	1	-1	-1	-1	-1	-1	1	1

Dataset with feature X and response Y.

We will now consider 3 weak classifiers (Hypotheses) H1, H2 and H3.
$H1: X \leq 0.35 \rightarrow Y = 1, else\ Y = -1$
$H2: X < 0.75 \rightarrow Y = -1, else\ Y = 1$
$H3: X \underline{\leq} 0.3\ or\ X \geq 0.95 \rightarrow Y = 1, else\ Y = -1$
Compute the weights of all the instances after the first round of boosting for each of the above mentioned weak classifiers. Use the initial set of (uniform) weights for each classifier. Do not perform normalization of the weights at the end of the iteration.

Problem 2. Consider an itemset $X = \{x_1, x_2, \dots, x_d\}$ with a support denoted by **[15 points]**

$$P(X) = \frac{\#transactions\ that\ contain\ X}{N},$$

where N is the total number of transactions. Suppose we measure the degree of association among items in the itemset with the following generalized cosine similarity measure:

$$cosine(X) = \frac{P(X)}{\sqrt{\prod_i P(x_i)}} = \frac{P(X)}{\sqrt{P(x_1)P(x_2)\cdots P(x_d)}}$$

For example, if $X = \{a, b\}$, then $cosine(\{a, b\}) = P(\{a, b\})/\sqrt{P(\{a\})P(\{b\})}$.

1) Express a simplified formula for the generalized cosine measure when all the items in X are independent.
2) Is the measure monotone, anti-monotone, or non-monotone when the size of itemset X is increased? In other words, when the set $X = \{a, b\}$ becomes $X' = \{a, b, c\}$, will the measure be non-decreasing (monotone), non-increasing (anti-monotone), or neither (non-monotone).

Problem 3. Apriori Algorithm **[20 points]**

Consider the following set of candidate 3-itemsets:

$\{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, c, d\}, \{a, c, e\}, \{a, c, f\}, \{a, d, e\}, \{b, c, d\}, \{b, c, e\}, \{b, c, f\},$
$\{b, d, e\}, \{c, d, e\}.$

1) Construct a hash tree for storing the above 3-itemsets. Assume the hash tree uses a hash function where items a, d are hashed to the left child of a node, items b, e are hashed to the middle child,

while items c, f are hashed to the right child. A candidate k-itemset is inserted into the tree by hashing on each successive item in the candidate and then following the appropriate branch of the tree according to the hash value. Once a leaf node is reached, the candidate is inserted based on one of the following conditions:

Condition 1: If the depth of the leaf node is equal to k (the root node is assumed to be at depth 0), then the candidate is added to the leaf node irrespective of the number of itemsets already stored at the node.

Condition 2: If the depth of the leaf node is less than k, then the candidate is added to the leaf node as long as the number of itemsets stored at the leaf node is less than or equal to $maxsize = 2$. Otherwise, change the leaf node into an internal node and distribute the candidates (including the new candidate to be added) to its children based on their respective hash values.

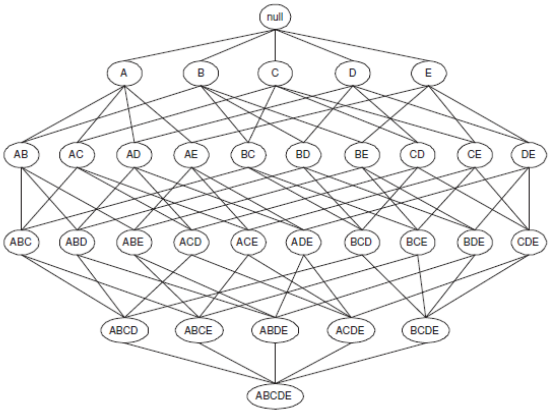
2) Consider a transaction that contains items $\{a, b, d, e, f\}$. Count the number of leaf nodes in the hash tree to which the transaction will be hashed into.

3) Suppose all the candidate 3-itemsets above are frequent. List all candidate 4-itemsets that can be generated from the frequent 3-itemsets using the candidate generation procedure for *Apriori*.

4) List all candidate 4-itemsets that survive the candidate pruning step of the *Apriori* algorithm.

5) Based on the list of frequent 3-itemsets given above, is it possible to generate a frequent 5-itemset? State your reason clearly.

Problem 4. Consider the lattice structure shown in the following figure. **[20 points]**



Search space for transaction database that contains 5 items.

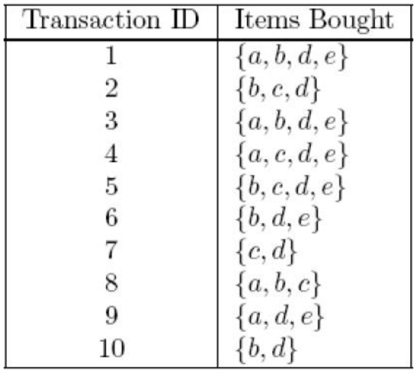
We are interested in finding all maximal and closed itemsets in a given data set.

1) What is the minimum and maximum number of maximal frequent itemsets one can generate from such a database?
2) What is the minimum and maximum number of closed frequent itemsets one can generate from such a database?

3) If all the transactions that contain itemset $\{A, B\}$ also contain items C and E, list all the itemsets that are guaranteed to be not closed.

4) If the support of itemsets $\{B\}$ and $\{B, C, D\}$ are identical, list all the itemsets that are guaranteed to be not closed.

Problem 5. Support and Confidence [15 points]



Transaction ID	Items Bought
1	$\{a, b, d, e\}$
2	$\{b, c, d\}$
3	$\{a, b, d, e\}$
4	$\{a, c, d, e\}$
5	$\{b, c, d, e\}$
6	$\{b, d, e\}$
7	$\{c, d\}$
8	$\{a, b, c\}$
9	$\{a, d, e\}$
10	$\{b, d\}$

The list of items bought in different transactions.

1) Draw a contingency table for each of the given rules using the transactions shown in the above table.
$\{b\} \rightarrow \{c\}$
$\{a\} \rightarrow \{d\}$
$\{b\} \rightarrow \{d\}$
$\{e\} \rightarrow \{c\}$
$\{c\} \rightarrow \{a\}$

2) Use the contingency tables in part 1) to compute and rank the rules in decreasing order according to support and confidence.

Problem 6. Text Classification [15 points]

Text classification is one of the most important tasks in Natural Language Processing. In this problem, you will use Python and Scikit-Learn package to write a simple program for text classification by following these steps:

1) Pre-process and load text data to your program.
Please use 20newsgroups dataset (http://qwone.com/~jason/20Newsgroups/).
If you are using scikit-learn package, you do not have to load the data manually.
2) Extract features from a text corpus.
This step will convert text to numerical vectors.
3) Split the dataset to training and testing sets by 80%/20%.
4) Use scikit-learn to train a classification model on the training data.
5) Use the trained model to predict class labels, e.g. sentiment, for samples in the testing data.
6) Use different metrics, such as Accuracy and F1-score, to evaluate your model.

Additional notes:
Jupyter notebook is recommended.
Include your codes and intermediate output of each step (Less than 3 pages) in your report.
If you use the public/open source codes, please provide the links to the sources.

Sources:

https://stackabuse.com/text-classification-with-python-and-scikit-learn/
https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a
https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f