CMPT 459 Fall 2020
Data Mining
Martin Ester
TAs: Madana Krishnan Vadakandara Krishnan and Rhea Rodrigues

Assignment 3

Total Marks: 100

Assignment 3.1 (40 Marks)

Mining frequent itemsets can be expensive. In a large, dynamic database of transactions, we can store the set of frequent itemsets and incrementally update that set upon arrival of a set of new transactions. Let *DB* denote the last state of our database and ΔDB a set of new transactions. The task is to incrementally determine the set of frequent itemsets in $DB \cup \Delta DB$ with respect to min-supp, without re-applying the Apriori-algorithm to the whole updated database $DB \cup \Delta DB$. More specifically, given the sets *DB* and ΔDB and the set of all frequent itemsets in *DB* together with their support, return the set of all frequent itemsets in $DB \cup \Delta DB$, without re-applying the Apriori-algorithm to the whole updated database $DB \cup \Delta DB$.

You can assume a (non-incremental) implementation of the Apriori-algorithm

Apriori (S: set of transactions, min-supp: float)

that returns all itemsets that are frequent in S together with their support. Note that min-supp is a relative frequency threshold.

a) Prove the following property: If an itemset is not frequent in *DB* and not frequent in ΔDB, then it cannot be frequent in $DB \cup \Delta DB$.

b) Based on the property that we have proven in a), as a first step, the incremental Apriori algorithm applies the non-incremental Apriori algorithm to ΔDB to determine the frequent itemsets in ΔDB and their support. After having performed this first step, for which itemsets do you need to count the support in *DB*, and for which itemsets do you need to count the support in ΔDB? Explain why the support counting is necessary.

Assignment 3.2 (60 Marks)

Implement the Local Outlier Factor (LOF) algorithm in python. For every point, compute the LOF value and determine whether the point is an outlier or not, based on a threshold that you have to choose. Your program shall read a file of two-dimensional points in CSV-format and produce a plot of the dataset, where the color of a point is red if the point is an outlier and blue otherwise.

You have to implement LOF "from scratch", i.e. you cannot use an existing LOF implementation, but you do not have to implement the plot function from scratch.

Apply your program on the outliers-3.csv dataset provided **here** (https://coursys.sfu.ca/2020fa-cmpt-459-d1/pages/Assignment_3_Datasets) and assume the data set is in the same directory as the program.

Your program should accept one input, the hyperparameter k.

In your PDF report:

- Explain how you set the hyperparameter k.
- Explain how you choose the threshold for determining whether a point is an outlier or not.
- Attach a screenshot of the LOF plot for outliers-3.csv data set.

Submission details

1. Submit a PDF Report for the solution to Assignment 3.1.
2. For Assignment 3.2:
    a. Submit a python program that takes one input parameter k.
    b. The required explanations and screenshot in the PDF report.