CS57300: Homework 5

Due date: Friday May 1, midnight (submit via turnin)
Note: Since this is due on the last day of classes, you cannot use late days on this assignment.

Association Rules

In this programming assignment you will implement an association rule algorithm and apply it to the *Yelp* data. Instructions below detail how to use turnin to submit your code and assignment on data.cs.purdue.edu. **Alternate submissions (i.e., outside the turnin system) will not be accepted.**

You will consider the following setup, unless otherwise specified:

- Data: Use the `stars_data.csv` data.
- Features: Use the binary word occurrence features computed for the top 201-2200 frequently occurring words.
- Class Label: Use the single class label ***isPositive***.

1. Implement the Apriori association rule algorithm. (25 pts)
   - Use the full data and 2001 binary features (2000 word features plus the class label).
   - Consider only frequent itemsets of size 1-3. Construct rules with a single consequent (e.g., `IF good AND service THEN isPositive`).
   - Use a support threshold of 5% and a confidence threshold of 25%.

   (a) Determine the size of the pattern space for itemsets (e.g., the number of elements in the lattice).

   (b) Computing the association rules that meet the minimum thresholds above.
   Track how many itemsets are:
   (i) considered by the algorithm (i.e., support is counted) and found to be frequent, and
   (ii) considered by the algorithm but found to be infrequent.

   (c) What is the pruning ratio of the Apriori algorithm in this case?
   Pruning ratio is defined as the percentage of itemsets not considered to be a candidate because (1) they are not generated during candidate generation or (2) they are pruned during the candidate pruning step.

   (d) What is the false alarm rate of the Apriori algorithm in this case?
   False alarm rate is defined as the percentage of candidate itemsets that are found to be infrequent after explicitly counting support.

   (e) Report the top 30 association rules that are discovered, ordered by confidence. State each rule along with its support and confidence values. Discuss whether the results are interesting or surprising based on your past experience analyzing this dataset.

2. Consider using the χ^2 score instead of confidence as the interestingness measure. (10 pts)

   (a) Describe an arbitrary contingency table you would use for calculating χ^2 from a rule in this context—where the rows correspond to whether the antecedent holds (or not) and the columns correspond to whether the consequent holds (or not).

   (b) Give the formula for calculating χ^2 from a rule in this context.

   (c) For one of your reported rules above, show the associated contingency table and calculated χ^2 score.

   (d) Describe how the numbers in the contingency table will change when you *generalize* the rule (e.g., by removing terms from the antecedent).

   (e) Describe how the numbers in the contingency table will change when you *specialize* the rule (e.g., by adding terms to the antecedent).

   (f) For your example contingency table, report the cell counts for the *best possible* specialization of your reported rule. What is the accuracy of the best possible specialized rule? What is the accuracy of your initial rule (from the contingency table above)?

   (g) What can we guarantee about the accuracy of further specializations of the rule? How can this be used to prune the space of rules during search? How does this relate to the Apriori principle?

3. Modify your association rule algorithm to use different search criteria (PDM p.439). (15 pts)

   (a) Use the χ^2 score as the interestingness measure and a test of significance ($\alpha = 0.05$) as the promise criterion.

   (b) Rerun your algorithm and report the top 30 newly found rules (ranked inversely by significance). Discuss the results and how they compare to the rules you identified in part 1.

   (c) Discuss how association rule algorithms can suffer from multiple comparison problems[1].

   (d) Outline how to use a Bonferroni correction to adjust for multiple comparison problems when using the χ^2 interestingness measure.

   (e) Rerun the algorithm using Bonferroni correction and discuss any change in the discovered rules.

[1]Reference: D. Jensen and P. Cohen (2000). "Multiple Comparisons in Induction Algorithms." *Machine Learning* 38: 309-338.

Submission Instructions:

After logging into data.cs.purdue.edu, please follow these steps to submit your assignment:

1. Make a directory named $'yourName\_yourSurname'$ and copy all of your files there.
2. While in the upper level directory (if the files are in /homes/neville/jennifer_neville, go to /homes/neville), execute the following command:

   turnin -c cs57300 -p HW5 your_folder_name

   (e.g. your prof would use: turnin -c cs57300 -p HW5 jennifer_neville to submit her work) Keep in mind that old submissions are overwritten with new ones whenever you execute this command.

   You can verify the contents of your submission by executing the following command:

   turnin -v -c cs57300 -p HW5

   Do not forget the -v flag here, as otherwise your submission would be replaced with an empty one.

Your submission should include the following files:

1. The source code in python.
2. Your evaluation & analysis in .pdf format. Note that your analysis should include learning curve graphs as well as a discussion of results.
3. A README file containing your name, instructions to run your code and anything you would like us to know about your program (like errors, special conditions, etc).