Programming project 1

October 1, 2012

Due date: Monday, October 15 2012 at 4pm.

Project logistics: You have the option to choose one of the two projects described below. A successful submission of your project will consist of three parts:

1. Source code of your implementation along with instructions of how to compile and run your program.
2. The input data you have used for your experiments in the right input format. For each input file you provide you should also submit the corresponding output file of your program. The files should be named so that if the input file has name `X.input` then the corresponding output file should be named `X.output`.
3. A document that describes your method and reports the experimental findings of your method on the input datasets. More specifically, the documentation should consist of four parts: (1) A formal (mathematical) description of your measure. (2) A detailed description of your algorithm, including pseudocode and running-time analysis. (3) An intuitive explanation why you have picked this measure and this algorithm. (4) A description of your experimental findings. The documentation part should not be more than *three (3)* pages long. The requirement is that once someone reads your writeup he/she should be able to reproduce your experiments.

Project submissions that do not contain *all* three parts will not be graded. You need to email the files of your project to me (`evimaria@cs.bu.edu`, by Monday October 15, 2012 at 4pm. The zip file should be named using your first and last name; i.e., `firstname_lastname.zip`.

Project 1: In class, we have discussed methods for reducing the size of the set of frequent itemsets output by a frequent-itemset mining algorithm. For example, we have discussed the notion of *maximal* and *closed* itemsets. We have also discussed how one can use the *Set Cover* and the *Maximal Coverage* problems to pick itemsets that cover all or as many of the itemsets in the frequent-itemset collection.

Devise your own measure of interestingness or goodness of an itemset or a collection of itemsets. Give an algorithm for finding the best k itemsets according to your measure. Use values of $k = 2, 4, 8, 16, 32, 64$. Implement your algorithm and report your experimental findings.

Resources: If your algorithm takes as input the set of frequent itemsets extracted by a standard itemset-mining algorithm you do not need to implement this algorithm yourself. Instead, you can use any of the existing implementations online. For example, look at: `http://www.adrem.ua.ac.be/~goethals/software/`

You can test your algorithms on datasets available on the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/datasets.html`). For the purpose of this project you should use at least two of the following datasets:

1. `Lenses:` (`http://archive.ics.uci.edu/ml/datasets/Lenses`)
2. `Mushroom:` (`http://archive.ics.uci.edu/ml/datasets/Mushroom`)
3. `Text datasets:` (`http://archive.ics.uci.edu/ml/datasets/Bag+of+Words`). For the text dataset you can use any one of the text datasets that are available on this webpage.

Note: You may need to preprocess the datasets (e.g., eliminate some features or some transactions that contain items with low frequency). In your documentation you need to describe in detail all the preprocessing you have performed on your data. You also need to provide the input files after your preprocessing so that we are able to execute your program.

Project 2: As we have discussed, different clustering objectives and different clustering algorithms. Each one of these algorithms when applied to the same dataset might give different results. In class, we have described the *disagreement distance* that measures how different two clusterings are.

Devise your own measure that computes the distance/similarity between the output of two different clustering algorithms. I.e., given two clusterings of the same dataset, your measure should be such that it takes small values if the clusterings are similar and large values if the clusterings are dissimilar. Ideally, two identical clusters should have distance 0. Describe your measure and its properties. Given an algorithm for evaluating the distance between two clusterings. Implement your algorithm and report your experimental findings.

Resources: For your experiments you do not need to implement the clustering algorithms yourself. For example, you can use any of the clustering algorithms that are already implemented in Matlab.

For this project you can test your algorithms on datasets available on the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/datasets.html`). For the purpose of this project you should use at least two of the following datasets:

1. `Wine:` (`http://archive.ics.uci.edu/ml/datasets/wine`)
2. `Communities and crime:` (`http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime`)
3. `Text datasets:` (`http://archive.ics.uci.edu/ml/datasets/Bag+of+Words`). For the text dataset you can use any one of the text datasets that are available on this webpage.

Note: You may need to preprocess the datasets (e.g., eliminate some features or some transactions that contain items with low frequency). In your documentation you need to describe in detail all the preprocessing you have performed on your data. You also need to provide the input files after your preprocessing so that we are able to execute your program.