University of Ioannina,
Department of Computer Science & Engineering

Data Mining – CSE 012/CS 059
Fall 2015

September Exam Assignment

The deadline for this Assignment is **September 27**, by the end of the day. Submit everything electronically, either through turn-in or email. Details for the turn-in, and how to write reports are on the Assignments web page of the course. For late submissions the late policy on the page of the course will be applied. There are no free passes for this assignment. Details for the oral exam will be posted after the assignments have been handed in.

Question 1

Assume that you are given as input a table with *n* rows and *m* columns, with 0/1 values. You want to find all *(r,c)-tiles* of 1's, that is, sets of *r* rows and *c* columns such that the corresponding submatrix has all 1's. Note that tiles may be overlapping. Describe an efficient algorithm for solving this problem that makes use of the APriori idea.

Question 2

Prove that for an undirected graph the stationary distribution of a random walk is proportional to the degree of the nodes. If P is the transition matrix of the random walk, and π is the stationary distribution for which $\pi = \pi \cdot P$, show that for node i the probability π_i is proportional to d_i where d_i is the number of edges incident on node i.

Question 3

In Yelp some businesses have a very large number of reviews, and it is hard for users to go over all of them to find out what they want. Therefore, we want for each business to select K reviews that capture all the **aspects** of the business as well as possible. Assume that there are $A_1, \dots, A_m$ aspects in total for all businesses that are known in advance (e.g., quality, location, price, service, etc). Also, through some preprocessing for each review, we know which aspects are being discussed in the review. Given N reviews for a business we want to select K of those so that in the final collection as many of the m aspects of the business as possible are being discussed in at least one review in the collection.

- Show that there is a greedy algorithm for the problem that has a constant approximation ratio with respect to the optimal algorithm that constructs the collection that contains the maximum number of aspects.
- What happens if we want all of the m aspects to appear in the collection?

Question 4

An important problem in online social networks is link prediction, that is, predicting future links between the network users. The algorithms for link prediction are used for friendship recommendations, so as to grow the network. In this question you will experiment with different algorithms for link prediction.

There are several approaches to this problem. You will consider two different approaches for this assignment.

In the first approach, for each node v, we compute a score for each link (v, u) between v and every node u that v is not already connected to. Then, we rank the possible new links for node v based on this score, and we keep the K first, for a constant K. You will consider three different methods for computing the score of a link (v, u).

1. The first method computes the number of common neighbors between v and u
2. The second method considers again the neighbors of v and u, but in this case each common neighbor z contributes a weight to the score function equal to $1/\log d_z$ where d_z is the degree of z.
3. The third method is the most complex one. For each node v, we perform a random walk with restarts, where the restart is always at node v. The score for the link (v, u) is the probability of the random walk being at node u in the stationary distribution.

The second approach views the link prediction problem as a classification problem, and constructs a classifier which for each pair (v, u) tries to predict if this link will appear in the network or not. For the classifier features we can use the scores we defined before, as well as other features based on the characteristics of the node, the behavior of the node in the system, or the network.

You will experiment with both approaches. You will use the data and the network from Question 3 in Assignment 4 (report again the characteristics of the dataset in your report). Select at random 100 nodes with at least 20 neighbors that will become your test set. These are the nodes for which we want to do link prediction. You will perform three different experiments:

1. In the first experiment, you will follow the first approach. For each of the nodes in the test set, remove one of the neighbors randomly. This is the link you want to predict. You will compute the score according to the three methods outlined above, and you will keep the K links with the highest scores for $K = 1,2,5,10$. Then, you will compute the precision of the method as the fraction of nodes for which the link that you removed is in the top K suggestions. (If there are ties, assume that the correct node is pushed to the top). For comparison, create a Random algorithm that selects K nodes at random. Create a plot with the mean (over the 100 nodes) precision, for these four methods.
   Bonus: Propose, implement, and test a different method for computing the score of a link.
2. In the second experiment you will create a classifier that uses features that you will define, and tries to predict for a pair (v, u) if a link will be created in the future. The classifier must have at least 10 features (you can also use information from the businesses to create features). In the training set you should not include the 100 nodes in the test set. For the test nodes, keep 10 of their neighbors, and for all the remaining possible links run the classifier to predict if the links will appear in the future. Test at least 2 classification methods, and report the average (over the 100 nodes) precision and recall of the classifiers.
3. Finally, you will consider a method that combines the two approaches. Build a Logistic Regression classifier for experiment 2, and use it to compute a probability for a node v and a potential link (v, u). Use these probabilities to rank the nodes and select the top K most likely links, and compute the same metrics as in experiment 1. Add an additional curve in the plot with the rest of the methods.

In all experiments report the average values over 10 different random selections of the 100 nodes in the test set.

Submit your code, and a report where you will describe the choices you made in the implementation, and the intuition behind these choices. You should do your own implementation of the random walk with restarts. In the report, comment also on the results. As always the report is very important for the evaluation of your assignment.