University of Ioannina,
Department of Computer Science & Engineering

Data Mining – CSE 012/CS 059
Fall 2014

Assignment 3

The deadline for this part is at the beginning of the class on January 13th. For late submissions the late policy on the page of the course will be applied. The details for the turn-in are on the page of the course. Turn in the requested material as well as all the code that you have written, and some instructions on how to run your code.

Question 1

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 2

For this question you will create a classifier for spam email. You will use the SpamAssassin dataset which you can find online (the link is on the class web page). To train the classifier, you need training examples which belong to the positive class (spam), and the negative class (non-spam), which you will download from the SpamAssasin site, and you will combine them into a single dataset (of at least 2000 examples). This dataset you will split into training (80%) and testing (20%) subsets.

Each example is the text of an email in a separate file. From this text, you will create features for your classifier. Some of the features should be words from the emails, for which the value of the feature will be the tf-idf value of the word. From the words you can select you can perform a selection of the ones you think are more important, e.g., the words with the largest weight, or the words in the subjects, etc. Beyond words, the features can be anything that you think can help in discriminating between spam and non-spam emails. For example, the size of the email, or the domain of the sender, etc. Combinations of the above are also possible. Your classifier should have at least 20 features (if you use all the words as features it will be a lot more). Each email becomes now a vector in the feature space.

Once you decide on the features you will extract them from the text, and you will create the input for training and testing. For the classification algorithm, you should use the LibLinear package (the link is on the class web page) which implements Logistic Regression classification (you would need to transform the input to what the program needs). After training, run the classifier you produced over the test set.

You should submit a report that explains the procedure above: What data you picked, how you split it into training and test subsets, what features you created, what algorithm you used, and the results of classifier on the test set. In the discussion of the results give the accuracy of the classifier, and the confusion matrix. You should also put a threshold θ on the probability of the positive class, and create the precision-recall curve for the values $\theta \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. Submit also your code and the input and output files, if you have changed their format.

Question 3

For the question you should implement the random walk with absorbing nodes described in class. Given a graph, and a set of absorbing nodes from the graph, for each non-absorbing node you should compute the probability that the random walk is absorbed at one of the absorbing nodes.

You will then apply the random on the graph that you will download from here. The nodes of the graph are blogs, and they are connected if one refers to the other. Also, the nodes belong to 39 different groups. Pick randomly 10 nodes from each group and make them absorbing. Then for each non-absorbing node compute the probability that it is absorbed in one of the 39 groups (the probability of being absorbed in a group is the sum of the probabilities of being absorbed at one of the 10 nodes of the group). Place the node to the group with the highest probability. Given the final placement , compute the confusion matrix, and the precision and recall for each group.

Turn in your code, and a report with your results and a comment on how well this method works for categorizing the blogs into groups.

Hint: Although you have 390 absorbing nodes, you essentially need to maintain only 39 probabilities.

Question 4

You work for TripAdvisor and you are in charge of managing the online reviews. Because each hotel has too many reviews you want to select K of them (where K is a small number, no more than 10) that capture all the **aspects** of the hotel as well as possible. Assume that there are $A_1, \dots, A_m$ aspects in total for all hotels that are known in advance (e.g., cleanliness, location, price, personel, etc). Also, through some preprocessing for each review, we know which aspects are being discussed in the review. Given N reviews for a hotel we want to select K of those so that in the final collection as many of the m aspects of the hotel 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 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?

Technical Details

For pre-processing, the following unix commands may come handy:

- **cut**: allows you to get specific columns from delimited data
- **sort**: sorts the rows of a file in lexicographic order, –n for numeric
- **uniq**: merges consecutive rows of a file that are identical.
- **grep**: finds a sting within a file

Do “man <command>” in unix/linux shell to get more information.