University of Ioannina,
Computer Science Department

Data Mining – CS 059
Fall 2012

Assignment 2

The assignment should be handed in at the beginning of the class on Tuesday November 20. For late submissions the late policy on the page of the course will be applied. The details for the turn-in will be announced on the page of the course.

Question 1 (Distance and Similarity)

1. Let x and y be two vectors of Euclidean length (L_2 norm) equal to 1. What is the relationship between the Euclidean distance $d(x, y)$ and the cosine similarity $cos(x, y)$ of the two vectors?
2. Let $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_n)$ be two vectors of size n. The sample Pearson correlation coefficient is defined as
$$\rho = \frac{\sum_{i=1}^{n}(x_i - \mu_X)(y_i - \mu_Y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_X)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_Y)^2}}$$
where μ_X, μ_Y are the mean values of vectors X and Y respectively.
Describe how we can use the cosine similarity to compute the sample Pearson correlation coefficient.
3. Let $U = \{u_1, \dots, u_n\}$ be a set of n items, and let R be an ordering (ranking) of the items. Define a "reasonable" distance function $d(R_1, R_2)$ between two rankings of the elements in U, which is a metric. Prove that your distance is a metric. The distance metric should have the property that the distance is maximized when one ranking is the inverse of the other.
4. Let $X = (x_1, x_2, \dots, x_n)$ be a vector of real numbers. For simplicity assume that the coordinates of the vector are ordered in decreasing order of absolute value, that is, $|x_1| \geq |x_2| \geq \cdots \geq |x_n|$. The p-norm of vector X is defined as $\|X\|_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p}$. Prove that when $p \to \infty$, $\|X\|_p \to |x_1|$.

Question 2 (Min-Hashing)

Do **Exercise 3.3.2** from the textbook Mining Massive Datasets by Anand Rajaraman and Jeff Ullman. Hand in the output of the intermediate steps of the Min-Hashing signature computation (as we did in class, and as it is done in the book), the final signature with all four functions, and the estimated and true similarity for all pairs of columns.

Question 3 (Min-Hashing and LSH)

In this exercise you will implement Min-Hashing and Locality Sensitive Hashing. You will test your implementation on the MovieLens 100k dataset which consists of a set of 943 users that have rated 1682 movies. . You can download the data from http://www.grouplens.org/node/73. Read the Readme file for details about the data, and process it as you need. For this exercise, we only care about the *set of movies* that a user has rated, and not the ratings. We want to compute the Jaccard similarity between the users

Compute the exact Jaccard similarity for all pairs of users and output the pairs of users that have similarity at least 0.5. Then compute the min-hash signatures for the users, and compute the approximate Jaccard similarity as we described in class. Use 50, 100, and 200 hash functions. For each value, output the pairs that have estimated similarity at least 0.5, and report the number of false positives and false negatives that you obtain. For the false positives and negatives, report the averages for 5 different runs.

Next, break up the signature table into b bands with r hash functions per band, as discussed in class, and implement Locality Sensitive Hashing. The goal is to find candidate pairs with similarity at least 0.6. Experiment with r=5, b =10 for the table with the 50 hash functions, r=5, b=20 for the table with the 100 hash functions, r = 5, b = 40 and r=10, b= 20 for the table with the 200 hash functions. Report the number of false positives and false negatives taking the average over 5 runs. How do these numbers change if we want similarity at least 0.8? What is the threshold of the sigmoid function?

You should turn in your code, the input file that you used, and the output files that you produced for the different runs. Also turn-in a file with the average numbers, and a report with your observations.

Technical details

1. For pre-processing the data, some unix commands that you may find useful are the following:
    a. cut: allows you to get specific columns from delimited data
    b. sort: sorts the rows of a file in lexicographic order, –n for numeric
    c. uniq: merges consecutive rows of a file that are identical.
2. Use the following hash function for the signatures:
    Select a large enough prime number R (e.g., R = 131071);
    Select a,b,c, random numbers in the interval [0,R]
    h(x) = (a*(x>>4) +b*x +c)%R;
3. For the implementation of LSH you do not need to implement a hash table or a list. Use existing implementations that come with the language (e.g., in C++ there is the STL implementation, check the web page of the object oriented programming course for more details).
4. Although it is not efficient, for simplicity you can instantiate the full user-movie 0/1 matrix. You will get bonus points for a more efficient implementation.