EE 380 L-10: Data Mining
Assignment # 2

This assignment has two parts to it. The first part is a practice with similarity measures. These measures, specifically called collaborative filters, are used to develop recommender systems. The second part is a practice with data reduction using PCA.

Part I – Collaborative Filtering and Recommender Systems.

Collaborative filtering is a known technique used for Recommender Systems. The main objective of recommender systems is to recommend Top-N items for a given user based on predicted ratings for items that have not been rated yet by the same given user. As an example, Netflix's movie recommendation can be implemented using collaborative filtering. In this assignment, the prediction of the ratings for these items is performed using collaborative filtering.

Collaborative filtering techniques can basically be divided into memory based techniques and model based techniques. In the following experiment, we will focus on memory based approaches, mainly, user-based collaborative filtering and item-based collaborative filtering. The first one utilizes the similarity computed between the active user and all other users whereas the other one makes use of the similarity available between two items. The similarity measures rely on the ratings available for two queried items. For instance, two users who gave close ratings to the same set of items will most likely have a similarity measure close to 1 whereas, two users who have different ratings for the same set of items are more likely to have similarity measure close to 0. The definition of similarity can be found through the Pearson correlation coefficient, the cosine similarity or the adjusted cosine similarity. An example of Pearson correlation coefficient is given in (1). For users *u and v, let* I_{uv} be the set of items rated by both users *u* and *v* with $r_{u,i}$ the rating of user *u* to item *i* and $\bar{r}_u$ is the average of ratings provided by user *u*.

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{v,i} - \bar{r}_v)^2}} \qquad (1)$$

Once the similarity measures are computed for all users, we can proceed by computing the missing ratings for the unrated items for each user. The predicted rating *r* is given in equation (2) based on a user-based approach where *v* belongs to the user in the neighborhood of the active user *u*. For this assignment, we will assume all other users are in the neighborhood of the active user.

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in N(u)} sim(u,v) * (r_{v,i} - \bar{r}_v)}{\sum_{v \in N(u)} |sim(u,v)|} \qquad (2)$$

In order to normalize the result, the mean rating of the user *v* is subtracted from his/her rating for item *"i"* and is divided by the sum of the absolute values of computed similarities in order to make sure that the predicted rating fall within the rating range, for example 1 to 5.

1. Objectives

A - You are asked to write a program in R to implement the above discussed memory-based collaborative filtering approach. The program will take as input:

- A text file name with the initial user-item matrix
- The number of ratings given in the text file.
- The number of users and the number of items,
- Name of output file to store the updated user-item matrix

The program will produce as output:

- The updated user-item matrix with all missing values, and stored in the chosen output file.

Use the provided text file: fileUI.txt, which includes ratings for certain items. The number of users is 10 and the number of items is 20. The number of available ratings is 50. The first column in the text file represents the user's ID, the second column represents the item's ID and the third column represents the rating that the specific user gave to the specific item.

B - Conduct a search on model-based collaborative filtering technique and write a brief summary of the technique.

2. Deliverables:

- Code showing the steps used for the above algorithm, and the similarity computations.
- Print out of updated user-item matrix.
- Description of model-based collaborative filtering.

Part II – Data Reduction/Transformation with Principal Component Analysis (PCA).

The objective of this part of the assignment is to practice data reduction/transformation with the use of principal component analysis (PCA), prior to the application machine learning models. In particular, this assignment will help you understand the general concept of PCA, interpret the output of the PCA MATLAB built-in function, and will set the stage for studying the impact of data reduction on classification accuracy.

1. Materials and Data

You are given an excel sheet (data-assignment-3.csv) that contains a dataset with 11 artificially-generated features (columns), in addition to a class feature that categorizes the 300 instances into either class A or B. Experiments will be conducted in MATLAB as specified in the procedure below.

2. Assignment

It is necessary to make sure you first understand the general concept of principal component analysis (PCA) in addition to its MATLAB function pca. You can refer to MATLAB help documentation (http://www.mathworks.com/help/stats/pca.html?refresh=true , or any other resource of your choice, then answer the following questions:

1. Explain briefly the general idea of principal component analysis (PCA) and its usefulness in data mining systems.
2. Upload the given dataset to MATLAB, and write a program with the steps explained in class to derive the principal components, and the related eigen values. You can use Matlab's eigen decomposition.
3. Verify your results, by using pca to obtain the principal components and their eigen values. Explain how the output of pca related to the elements described in class. What do the terms "coeff", "score" and the "latent vector" represent?
   a. What are the mathematical expressions (equations) that are used to obtain the score matrix from the original data?
4. Note that the components (columns) output from pca are already ordered in decreasing order. What is the significance of the first column of the coefficients matrix?
5. Each of the resulting principal components (PCs) has its own contribution to the variance. Plot the cumulative contribution of the PCs to the variance. In other words, plot the variance of PC_1, $PC_{1\text{-}2}$, $PC_{1\text{-}2\text{-}3}$, and so on. Hint: use the following normalized variance formula for normalizing each variance:

$$var(PC_i) = \frac{latent\ [i]}{\sum_{j=1}^{n} latent\ [j]}$$

where n is the total number of PCs. Note that the latent[i] already comprehends the square needed to measure energy, since these are the eigen values of the covariance matrix.

6. Based on the previous steps (2, 4, and 5), what feature selection strategy would you recommend for reduction or no reduction? If yes, which features should be reduced? Justify your answer.

4. Lab Deliverables

To accomplish this assignment, you are expected to submit:

- Matlab Code showing the steps to derive principal components
- Matlab code showing the use of pca
- A report with the answers to the above questions.