**Problem 1: Collaborative Filtering on Netflix Ratings**

The code for the assignment is written in c# as a 64-bit console application and is submitted as a zip file.   
The code is also uploaded in my github repository and can be found here <https://github.com/nivetha27/Machine-Learning-Assignments/tree/master/Assignment%202/Code/Assignment2>  
All the output files are in the same directory as the .exe file and can be found here <https://github.com/nivetha27/Machine-Learning-Assignments/tree/master/Assignment%202/Code/Assignment2/bin/x64/Debug>  
To get movie predictions for oneself, update only the movie ids and ratings in myTrainingRatings.txt and make sure the userId-1181332- is not updated. Run “Assignment2.exe true” from cmd.  
To see results of testingratings.txt, run “Assignment2.exe” from cmd.

Here are the key functions in the code :

1. Loading training data:
   * public static void loadTrainingDataIntoDict(string fullFileName)
   * This function loads the training data into user and movie dictionaries that are used for further processing
2. Processing training data:
   * public static void loadMeanUserVoteDataIntoDict()
   * This function computes the average rating for each user based on the training data
   * public static void getWeightForUsers()
   * This function computes the weight between all the user pairs and loads it in weights.txt file and in an array. In the case the file is already present, it reads from weights.txt and populates the array.
   * public static double calcWeight(int userIdA, int userIdI, Dictionary<int, double> movieOfUserADict, Dictionary<int, double> movieOfUserIDict)
   * This calculates the pearson coefficient between the given two users.
3. Evaluate test data:
   * public static void evaluateTestData(string fullFileName, double threshold)
   * Loads test data from the test data set file and gets the predicted rating for each test data point, eventually computes the mean absolute error and root mean square error.
   * public static double predictVote(int movieId, int userId, double threshold)
   * Computes the predicted vote for a given user and a movie, eliminates weights which are less than given threshold

In my algorithm for predicting ratings for a given user and a movie, I am including only weights that are more than a given threshold. This approach allows to consider only users who are closer/similar to the user for who the prediction is currently being computed. Below are my results for different thresholds.

|  |  |  |
| --- | --- | --- |
| Threshold | Mean Absolute Error | Root Mean Squared Error |
| 0 | 0.694711599217723 | 0.8864743924077412 |
| 0.05 | 0.694358374885737 | 0.8860496889334396 |
| 0.1 | 0.693315161730442 | 0.8848123687336344 |
| 0.15 | 0.691789457178226 | 0.8830380685988985 |
| 0.2 | 0.689842589893048 | 0.8807062888464707 |
| 0.25 | 0.687874698086059 | 0.8784809791118417 |
| 0.3 | 0.685989486672359 | 0.8764555440692968 |
| **0.35** | **0.684754580994746** | **0.8751945859961372** |
| 0.4 | 0.684760005592665 | 0.875830141619711 |
| 0.45 | 0.687613892702176 | 0.8809033568887424 |
| 0.5 | 0.693577481052001 | 0.8897412643196847 |

The least and therefore the best RMSE and MAE are for **threshold 0.35**.

**RMSE** is **0.684754580994746** and **MAE** is **0.8751945859961372**

**Shortcomings of the collaborative filtering algorithm**

* This algorithm is computationally intensive as it is linearly dependent on the number of users and movies. This will cause scaling problems for large datasets.
* For a new user who hasn’t rated any movie or a new movie for which there is no rating so far, this algorithm cannot make recommendations for such a user or a movie. In this case, we need to get more features like movie genre into training data and make prediction using these additional features.

**Extra-Credit**

I ran my collaborative filtering algorithm using the best threshold value 0.35 for myself by adding into the training set ratings for movies that I had watched before and assigned myself the user id - 1181332  
  
The movies I had rated and had added to the training data can be found here - https://github.com/nivetha27/Machine-Learning-Assignments/tree/master/Assignment%202/Code/Assignment2/bin/x64/Debug/myTrainingRatings.txt

The top 5 recommendations in the order of decreasing predicted ratings makes sense to me:

#1 - Raging Bull: Collector's Edition: Bonus Material  
#2 - The Peacekeeper Wars: Bonus Material  
#3 - Ghost in the Shell: Stand Alone Complex: 2nd Gig  
#4 - Finding Nemo (Full-screen)  
#5 - The Incredibles

#1 is relevant to me as I had given 5 rating for WWE: Armageddon(69), WWE: Royal Rumble(91)  
#2 seems to be fantasy adventure like lord of the rings(13) for which I rated 5 and also similar to the legend(281) in terms of war which I had rated 4  
#3 is similar to dragon ball(263), a Japanese anime, for which I had rated 5  
#4,#5 is also relevant as I had rated animation movies like Stuart Little 2(252) as 5, The Powerpuff Girls Movie(84) as 4.

**2.1** You are managing an intern hoping to launch a product, and you know the following:

* If the intern writes good code and has enough tests, the launch is more likely to be successful.
* If coffee is available, the intern writes better code.
* If coffee is available, the intern is more likely to write tests.
* Having a tight launch deadline results in the intern writing fewer tests

Please answer the following questions.

1. Draw the Bayesian network consistent with the statements above using variables *LaunchSuccess*, *CodeQuality*, *TestCoverage*, *Deadline*, and *HasCoffee*.
2. According to this network, if the intern writes tests, is the code more likely to be good? Why or why not?   
   In this case, since its known that intern writes test, it can be that intern has had coffee in which case the code quality is more likely to be good or it can be that there is a tight deadline that does not give any information about the code quality
3. According to this network, are the code quality and deadline independent? Why or why not?   
   No, they are not independent. *Code quality* and *deadline* become independent only given code quality’s parents which is *has coffee*.
4. According to this network, are the code quality and deadline independent given a successful launch? Why or why not?   
   No, they are not independent given successful launch as observing one factor may become less likely than the other given successful launch.
5. According to this network, are the code quality and deadline independent given that the intern did not get any coffee? Why or why not?   
   Yes they are independent, in this case as it becomes clear that the launch success is determined by deadline only.

**2.2** Suppose you have 4 binary variables: a, b, c, and d. Consider the Bayesian network structured as follows a→b→c→d. The network has the following parameters:

* P(a)=0.6
* P(b|a)=0.9
* P(b|¬a)=0.2
* P(c|b)=0.3
* P(c|¬b)=0.7
* P(d|c)=0.4
* P(d|¬c)=0.7

Please answer the following questions:

1. Compute the probability of P(d). (hint: you need to marginalize out the probabilities of a, b, and, c)

P(d) = ∑P(a,b,c).P(d)   
 = P(a,b,c,d) + P(¬a, ¬b, ¬c,d) + P(¬a, ¬b,c,d) + P(¬a,b, ¬c,d) + P(¬a,b,c,d) +P(a, ¬b, ¬c,d) + P(a, ¬b,c,d) + P(a,b, ¬c,d)  
 = 0.0648 + 0.0672 + 0.0896 + 0.0392 + 0.0096 + 0.0126 + 0.0168 + 0.2646  
 = 0.5644

P(a,b,c,d) = P(a)P(b|a)P(c|b)P(d|c) = 0.6 \* 0.9 \* 0.3 \* 0.4 = 0.0648

P(¬a, ¬b, ¬c,d) = P(¬a)P(¬b|¬a)P(¬c|¬b)P(d|¬c) = 0.4 \* 0.8 \* 0.3 \* 0.7 = 0.0672  
  
Similarly, computed for other cases.

1. How many summations did you need to compute? Can you do better with a different ordering of summations?  
   I needed 8 sums. Yes, it can be made better with only 3 sums as shown below.

P(b) = P(b|a)P(a) + P(b|¬a)P(¬a) = 0.9\*0.6 + 0.2\*0.4 = 0.62 , P(¬b) = 0.38

P(c) = P(c|b)P(b) + P(c|¬b)P(¬b) = 0.3\*0.62 + 0.7\*0.38 = 0.452, P(¬c) = 0.548

P(d) = P(d|c)P(c) + P(d|¬c)P(¬c) = 0.4\*0.452 + 0.7\*0.548 = 0.5644

1. Suppose we have a Bayesian network with a similar structure but with n binary variables in a chain instead, i.e. x1→x2→x3→…→xn. What is the fewest number of sums needed to compute P(xn)?   
   I would need (n-1) sums
2. What is the worst-case number of sums needed to compute P(xn)?   
   2(n-1) sums

**2.3** Consider again the above chain-structured Bayesian network, a→b→c→d. Suppose you have a training set composed of the following examples in the form (a,b,c,d), with "?" indicating a missing value: (0,1,1,1), (1,1,0,0), (1,0,0,0), (1,0,1,1), and (1,?,0,1). Show the first iteration of EM algorithm (initial parameters, E-step, M-step), assuming the parameters are initialized ignoring missing values.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | A | B | C | D |
| 1 | 0 | 1 | 1 | 1 |
| 2 | 1 | 1 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 |
| 4 | 1 | 0 | 1 | 1 |
| 5 | 1 | ? | 0 | 1 |

* P(a) = 4/5
* P(b|a) = 1/3
* P(b|¬a) = 1
* P(c|b) = ½
* P(c|¬b) = ½
* P(d|c) = 1
* P(d|¬c) = 1/3

**E-step**

P(b|a, ¬c,d) = P(a,b, ¬c,d)/ P(a, ¬c,d) = P(a,b, ¬c,d)/(P(a,b, ¬c,d) + P(a, ¬b, ¬c,d))

= P(a)P(b|a)P(¬c|b)P(d|¬c) / (P(a)P(b|a)P(¬c|b)P(d|¬c) + P(a)P(¬b|a)P(¬c|¬b)P(d|¬c)

= (0.8 \* 1/3 \* 0.5 \* 1/3) / (0.8\*1/3\*0.5\*1/3 + 0.8\*2/3\*0.5\*1/3) = (0.4/9)/(0.4/9 + 0.8/9)

P(b|a, ¬c,d) = 0.4/1.2 = 1/3

**P(b|a,** ¬**c,d) = 1/3  
P(**¬**b|a,** ¬**c,d) = 2/3**

**M-step**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | A | B | C | D | Expected Probability |
| 1 | 0 | 1 | 1 | 1 |  |
| 2 | 1 | 1 | 0 | 0 |  |
| 3 | 1 | 0 | 0 | 0 |  |
| 4 | 1 | 0 | 1 | 1 |  |
| 5 | 1 | 0 | 0 | 1 | 2/3 |
| 6 | 1 | 1 | 0 | 1 | 1/3 |

* **P(a)** = (3 + 1/3 + 2/3) / (4 + 1/3 +2/3) = **4/5**
* **P(b|a)** = (1 + 1/3) / (3 + 1/3 + 2/3) = (4/3)/4 = **1/3**
* **P(b|**¬**a)** = **1**
* **P(c|b)** = 1/ (2 + 1/3) = 1/(7/3) = **3/7**
* **P(c|**¬**b)** = 1/(2 + 2/3) = 1/(8/3) = **3/8**
* **P(d|c)** = **1**
* **P(d|**¬**c)** = (1/3 + 2/3)/(2 + 1/3 + 2/3) = **1/3**