**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> and all the output files are in https://github.com/nivetha27/Machine-Learning-Assignments/tree/master/Assignment%202/Code/Assignment2/bin/Debug

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 also 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, I am including only weights that are more than a given threshold during computation of predicted rating for a given user and a movie. This approach allows to consider only users who are closer/similar to the user for who the prediction is currently computed. Below are my results for different threshold.

|  |  |  |
| --- | --- | --- |
| 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 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.

**Extra-Credit**

I ran my collaborative filtering algorithm using the best threshold value 0.35.

**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? We cannot infer code quality from the test coverage as code quality and test coverage are conditionally independent given has coffee
3. According to this network, are the code quality and deadline independent? Why or why not? Yes they are independent. Code Quality is conditionally dependent only on *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 dependent given successful launch as observing one factor may become less likely than the other given the descendant.
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 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

1. How many summations did you need to compute? Can you do better with a different ordering of summations? Needed 8 sums
2. 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 2(n-1) sums
3. 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. Note that this problem does not use the parameters described in problem 2.2.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | D |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 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)

= 0.4/1.2

= 1/3