Recommender Systems

TOTAL POINTS 5

1. Suppose you run a bookstore, and have ratings (1 to 5 stars) of books. Your collaborative filtering algorithm has learned a parameter vector $\theta^{(j)}$ for user j, and a feature vector $x^{(i)}$ for each book. You would like to compute the "training error", meaning the average squared error of your system's predictions on all the ratings that you have gotten from your users. Which of these are correct ways of doing so (check all that apply)? For this problem, let m be the total number of ratings you have gotten from your users. (Another way of saying this is that $m = \sum_{i=1}^{n_m}\sum_{j=1}^{n_u} r(i,j)$). [Hint: Two of the four options below are correct.]

1 point

☑ $\frac{1}{m}\sum_{(i,j):r(i,j)=1}\sum_{k=1}^{n}((\theta^{(j)})_k x_k^{(i)} - y^{(i,j)})^2$

□ $\frac{1}{m}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}(\sum_{k=1}^{n}(\theta^{(j)})_k x_k^{(i)} - y^{(i,j)})^2$

□ $\frac{1}{m}\sum_{(i,j):r(i,j)=1}(\sum_{k=1}^{n}(\theta^{(j)})_k x_k^{(i)} - y^{(i,j)})^2$

☑ $\frac{1}{m}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}((\theta^{(j)})_i x_j^{(i)} - y^{(i,j)})^2$

2. In which of the following situations will a collaborative filtering system be the most appropriate learning algorithm (compared to linear or logistic regression)?

1 point

□ You've written a piece of software that has downloaded news articles from many news websites. In your system, you also keep track of which articles you personally like vs. dislike, and the system also stores away features of these articles (e.g., word counts, name of author). Using this information, you want to build a system to try to find additional new articles that you personally will like.

□ You manage an online bookstore and you have the book ratings from many users. You want to learn to predict the expected sales volume (number of books sold) as a function of the average rating of a book.

☑ You manage an online bookstore and you have the book ratings from many users. For each user, you want to recommend other books she will enjoy, based on her own ratings and the ratings of other users.

☑ You run an online news aggregator, and for every user, you know some subset of articles that the user likes and some different subset that the user dislikes. You'd want to use this to find other articles that the user likes.

3. You run a movie empire, and want to build a movie recommendation system based on collaborative filtering. There were three popular review websites (which we'll call A, B and C) which users to go to rate movies, and you have just acquired all three companies that run these websites. You'd like to merge the three companies' datasets together to build a single/unified system. On website A, users rank a movie as having 1 through 5 stars. On website B, users rank on a scale of 1 - 10, and decimal values (e.g., 7.5) are allowed. On website C, the ratings are from 1 to 100. You also have enough information to identify users/movies on one website with users/movies on a different website. Which of the following statements is true?

1 point

○ It is not possible to combine these websites' data. You must build three separate recommendation systems.

○ Assuming that there is at least one movie/user in one database that doesn't also appear in a second database, there is no sound way to merge the datasets, because of the missing data.

● You can merge the three datasets into one, but you should first normalize each dataset's ratings (say rescale each dataset's ratings to a 1-100 range).

○ You can combine all three training sets into one without any modification and expect high performance from a recommendation system.

4. Which of the following are true of collaborative filtering systems? Check all that apply.

1 point

□ For collaborative filtering, the optimization algorithm you should use is gradient descent. In particular, you cannot use more advanced optimization algorithms (L-BFGS/conjugate gradient/etc.) for collaborative filtering, since you have to solve for both the $x^{(i)}$'s and $\theta^{(j)}$'s simultaneously.

☐ Suppose you are writing a recommender system to predict a user's book preferences. In order to build such a system, you need that user to rate all the other books in your training set.

☑ For collaborative filtering, it is possible to use one of the advanced optimization algoirthms (L-BFGS/conjugate gradient/etc.) to solve for both the $x^{(i)}$'s and $\theta^{(j)}$'s simultaneously.

☑ Even if each user has rated only a small fraction of all of your products (so $r(i,j) = 0$ for the vast majority of (i,j) pairs), you can still build a recommender system by using collaborative filtering.

5. Suppose you have two matrices A and B, where A is 5x3 and B is 3x5. Their product is $C = AB$, a 5x5 matrix. Furthermore, you have a 5x5 matrix R where every entry is 0 or 1. You want to find the sum of all elements $C(i,j)$ for which the corresponding $R(i,j)$is 1, and ignore all elements $C(i,j)$ where $R(i,j) = 0$. One way to do so is the following code:

1 point

```
C = A * B;
total = 0;
for i = 1:5
  for j = 1:5
    if (R(i,j) == 1)
      total = total + C(i,j);
    end
  end
end
```

Which of the following pieces of Octave code will also correctly compute this total? Check all that apply. Assume all options are in code.

☑ total = sum(sum((A * B) .* R))

☐ C = A * B; total = sum(sum(C(R == 1)));

☐ C = (A * B) * R; total = sum(C(:));

☐ total = sum(sum(A(R == 1) * B(R == 1));

☐ I, **Hassan Rasheed**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

Learn more about Coursera's Honor Code