Feedback — XVI. Recommender Systems

Help

You submitted this quiz on **Mon 19 May 2014 10:06 AM IST**. You got a score of **4.00** out of **5.00**. You can attempt again in 10 minutes.

Question 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.]

Your Answer		Score	Explanation
☐ $\frac{1}{m} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} (\sum_{k=1}^{n} (\theta^{(k)})_j x_i^{(k)} - y^{(i,j)})^2$	✔	0.25	This incorrectly indexes into $\theta^{(j)}$ and $x^{(i)}$.
☐ $\frac{1}{m} \sum_{(i,j):r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - r(i,j))^2$	✔	0.25	This incorrectly used $r(i,j)$ as the actual rating.
☑ $\frac{1}{m} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} (\sum_{k=1}^{n} (\theta^{(j)})_k x_k^{(i)} - y^{(i,j)})^2$	✔	0.25	This correctly sums over all ratings and computes the predicted rating with the explicit sum $\sum_{k=1}^{n} \theta^{(j)})_k x_k^{(i)}$.
☑ $\frac{1}{m} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} (\sum_{k=1}^{n} (\theta^{(j)})_k x_k^{(i)} - y^{(i,j)})^2$	✔	0.25	This correctly sums over all ratings and computes the predicted rating with the explicit sum $\sum_{k=1}^{n} \theta^{(j)})_k x_k^{(i)}$.
Total		1.00 / 1.00	

Question 2

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

Your Answer		Score	Explanation
☐ 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.	✔	0.25	This system uses predetermined features and has only one user, so it is not a good application of collaborative filtering.
☑ You own a clothing store that sells many styles and brands of jeans. You have collected reviews of the different styles and brands from frequent shoppers, and you want to use these reviews to offer those shoppers discounts on the jeans you think they are most likely to purchase	✔	0.25	By combining the reviews of all the different shoppers, you can use collaborative filtering to find similar pairs that they might like.
☑ 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.	✔	0.25	This is a good application of collaborative filtering, as you can use the like / dislike as a rating to learn features for the articles and recommend articles similar to those each user likes.
☐ You manage an online bookstore and you have the book ratings from	✔	0.25	This is a regression problem of predicting sales volume from ratings data, so collaborative filtering is not applicable.

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.	
Total	1.00 / 1.00

Question 3

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:

```
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.

Your Answer		Score	Explanation
☑ total = sum(sum(A(R == 1) * B(R == 1));	✖	0.00	You cannot use R to perform logical indexing into A and B, since R does not have the same dimension as those two matrices.
☐ C = A * B; total = sum(sum(C(R == 1)));	✖	0.00	This sums up all the elements in `C(R == 1)`, where the "logical indexing" expression selects only elements of C whose index matches an index in R for 1 elements.
☐ total = sum(sum((A * B) * R));	✔	0.25	Multiplying `(A * B) * R` will perform regular matrix multiplication and won't "mask out" entries.

☑ total = sum(sum((A * B) .* R));	✔	0.25	This sums up all elements of `(A * B) .* R`, where the .* operator performs element-wise multiplication, setting the elements of `A * B` to zero that correspond to zero entries in `R`.
Total		0.50 / 1.00	

Question 4

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?

Your Answer		Score	Explanation
○ You can combine all three training sets into one without any modification and expect high performance from a recommendation system.			
◉ 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 0-1 range).	✔	1.00	By normalizing each dataset, you ensure that all ratings are on the same scale, so they are comparable during training.
○ You can combine all three training sets into one as long as your perform mean normalization and feature scaling **after** you merge the data.			
○ It is not possible to combine these websites' data. You must build three separate recommendation systems.			
Total		1.00 / 1.00	

Question 5

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

Your Answer		Score	Explanation
☐ 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.	✔	0.25	You can compute the cost function and gradient, so any of the advanced optimization algorithms will also work.
☑ When using gradient descent to train a collaborative filtering system, it is okay to initialize all the parameters ($x^{(i)}$ and $\theta^{(j)}$) to zero.	✖	0.00	You need to initialize them to different values so that you learn different features and parameters (i.e., perform symmetry breaking).
☑ Even if you 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.	✔	0.25	The collaborative algorithm can still leverage the ratings that are present to build a reasonable recommender system.
☐ Recall that the cost function for the content-based recommendation system is $J(\theta) = \frac{1}{2}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}\left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}($. Suppose there is only one user and he has rated every movie in the training set. This implies that $n_u = 1$ and $r(i,j) = 1$ for every i,j. In this case, the cost function $J(\theta)$ is equivalent to the one used for regularized linear regression.	✖	0.00	In this case, the cost function is just a sum of squared differences between a prediction

		$\theta^T x$ and true vaue y; this is exactly linear regression.
Total	0.50 / 1.00	