Recommender Systems

TOTAL POINTS 5

---

1. Suppose you run a bookstore, and have ratings (1 to 5 stars) **1 point**

    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.]

    ☐ $\frac{1}{m} \sum_{(i,j):r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - r(i,j) \right)^2$

2. In which of the following situations will a collaborative filtering system be the most appropriate learning algorithm **1 point**
    (compared to linear or logistic regression)?

    ☐ You're an artist and hand-paint portraits for your clients. Each client gets a different portrait (of themselves) and
    gives you 1-5 star rating feedback, and each client purchases at most 1 portrait. You'd like to predict what rating
    your next customer will give you.

    ☐ 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 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

    ☑ You run an online bookstore and collect the ratings of many users. You want to use this to identify what books are
    "similar" to each other (i.e., if one user likes a certain book, what are other books that she might also like?)

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

    ○ 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).

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

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.

    ☑ 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.

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. **1 point**
    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;
      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.

- [x] total = sum(sum((A * B) .* R))

- [x] 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));