✖

# Final Exam - Part 3

12 questions

1.
What is the purpose of decision-support metrics such as reversals, precision, or ROC?

- (selected) They’re designed to measure how effectively a recommender can be used to distinguish between desirable and undesirable items.
- They’re designed to measure the total amount of error in the predictions given by a recommender.
- They’re designed to measure whether the top-recommended items are indeed the best items available.
- They’re designed to measure what percentage of items in the product set users actually like.

2.
Which of these statements best explains how we perform an n-fold cross validation for getting a more accurate measure of the accuracy experienced by users in a recommender system?

- Divide the data set into n partitions of **items**; hold one partition out as test data and train the recommender on the other partitions. Now measure the accuracy of prediction for the withheld items for each user and average them.

- Divide the data set into n partitions of **users**; hold one partition out as test data; train the recommender on the other partitions. Divide the withheld user data into “query” data used for training and “test” data. Measure the accuracy of predicting the test data from the query data for each user and average.
- Randomly withhold n ratings from the dataset. Predict each rating from all other data, and average the results.
- Pick a random set of test ratings. Divide the remaining ratings into n batches of test data. Train the recommender separately on each batch, and predict the test ratings with each trained recommender. Average the results.

---

3.
Why would you want to conduct a field experiment or A/B tests of a recommender system?

- To compare the accuracy or rank metrics across different algorithm variants.
- To see how much metrics for accuracy or rank vary depending on the user for whom predictions and recommendations are being generated.
- To study user’s subjective opinions of a recommender systems they’ve been using.
- To measure the effects of different recommenders on user behavior.

---

4.
What do time-averaged metrics let us do?

- Estimate more accurately the amount of time it takes for the recommender to provide recommendations, averaging the expensive start-up phase with the more efficient steady-state.
- Measure how recommender performance changes as the system grows and matures.
- Average out and ignore temporal effects in recommendation.
- Estimate user retention by discounting recommendations given to users who would have abandoned the system due to earlier poor performance.

---

5.

When using item-item CF with unary data, we usually just sum the similarities between the item and its neighbors, rather than computing a weighted average. Why?

- Since there are no ratings, the weighted average is effectively an average of a set of 1s, which is always 1. Summing similarities creates a meaningful score.
- Because summation is significantly faster computationally than computing a weighted average, and a major benefit of item-item is faster performance.
- Because weighted averages cannot be pre-computed, but sums can be easily cached and reused for future computations.
- Because sums help adjust for the fact that we don't really know whether the non-ratings represent items that are disliked or just never consumed.



---

6.
Which of the following is a downside of FunkSVD *compared to traditional SVD approaches*?

- (selected) The resulting user-/item-feature matrices are not orthogonal.
- It ignores missing values
- It takes a lot more time.
- The dimensions of the resulting matrix are not easily understandable.

---

7.
Sometimes item similarity vectors are normalized to the unit vector (so the squares of the similarities of an item's neighbors sum to 1). What does this do?

- (selected) Weights items with few neighbors more highly, as their similarities likely contain more information.
- Weights users with many ratings more highly, as they are likely to contain more preference information.
- Weights users with few ratings more highly, as their preferences are more focused.
- Devalues extreme ratings, to help balance out raters who express their ratings only in terms of extremes.

---

8.
What are the update rules in each iteration of Funk-SVD derived from?

- The partial derivative of the prediction error of SVD-based prediction with respect to the particular parameter to be updated.
- The partial derivative of the rank accuracy of SVD-based recommendation with respect to the particular parameter to be updated.
- A random movement to an adjacent parameter value in hopes that it will reduce the error.
- Selecting the parameter value that will minimize the expected RMSE of predicting ratings with the computed model.

---

9.

What is the basic idea behind matrix-factorization recommender algorithms such as SVD?

- To mathematically simplify collaborative filtering by grouping all of the similar neighbors together.
- To speed up the process of building models for collaborative filtering by avoiding the expensive correlation computations.
- To transform the user-by-item ratings space into user-by-feature preference and item-by-feature expression for some set of latent features
- To incorporate content attributes into ratings-based collaborative filtering.

---

10.

In Burke's paper and interview on hybrids, he referred to several types of hybrids including switching and cascade hybrids. What are these?

- Switching hybrids select the prediction or recommendation from one underlying recommender (whichever is expected to be best); cascade hybrids perform a first-round recommender to get a coarse approximate recommendation set and then feed that into a second-round recommender to refine/filter/reorder the recommendations.

- Switching hybrids combine the results from multiple recommenders in a weighted average; cascade hybrids perform a first-round recommender to get a coarse approximate recommendation set and then feed that into a second-round recommender to refine/filter/reorder the recommendations.

- Switching hybrids select the prediction or recommendation from one underlying recommender (whichever is expected to be best); cascade hybrids run several recommenders in parallel and choose the recommendation that has the highest confidence score.

- Switching hybrids combine the results from multiple recommenders in a weighted average; cascade hybrids run several recommenders in parallel and choose the recommendation that has the highest confidence score.

---

11.

What is the interpretation of each dimension of the resultant matrices in an SVD-based recommender system.

- Each dimension represents a set of expressed item attributes (e.g., cast members, genre, etc.) over which users have a set of preferences. It is like content filtering, but more automatic.

- Each dimension represents one of the user's item ratings; when the matrices are multiplied, the ratings get attached to the relevant products.
- Each dimension represents a cluster of users with a particular preference centroid.
- Each dimension represents a latent orthogonal taste dimension--usually not easily interpreted in terms of either item or user attributes

---

12.

MusicFX picked radio stations to play in a shared company gym. Users rated each station (genre). One early feature of the system was that it would avoid playing any station that any single person in the gym marked as hated. What was the problem they experienced that led them to change this feature?

- They quickly discovered that too often everything was hated by someone, and couldn't find any stations to play.
- They discovered that people didn't want to be seen as imposing their tastes, so they were reluctant to admit that they hated specific station.
- They discovered that some people would mark a station as "hated" even if they just mildly disliked it, and sometimes just to force the system to change the station.
- The feature worked well, but they had to discontinue using it because they had a contract that required them to play a wider variety of music.

---

Submit Quiz

(https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc)