



Evaluate Recommender Systems

Note

- The #1 metric is how real customers in the real world react to the recommendations you produce

Online Evaluation

- Expensive
- Time consuming
- User exhausting

Offline Evaluation

- train/test split
- K-fold cross-validation

Offline Metrics

Offline* metrics

- ***we are predicting on historical data, not future data**
 - Only possible through online A/B tests

Mean absolute error

- Difference between actual rating and predicted rating
- Lower MAE means a better model

Root mean squared error

- Similar to MAE
- Penalizes the term more when the error is high

Hit rate

- Hits in test / number of users
- Better alternative to MAE or RMSE

Hit rate

- Generate top N recommendations for all users in our test dataset
- If top N contain something the users rated = 1 hit

Hit rate

- **Measure with Leave One Out
Cross-Validation**

Leave One Out Cross-Validation

- Compute top N recommendations for each user in training data
- Intentionally remove 1 of the items from the user's training data
- Test the recommended ability to recommend that removed item

Leave One Out Cross-Validation

- Hit rate is very smaller
- Hard to measure without a large dataset

Average reciprocal hit rank

- Variation of hit rate
- Add reciprocals of the rank of each hit
- Accounts for where in the top N lists the hits appear

Average reciprocal hit rank

- More credit for successfully recommending item in top slot than in bottom slot
- Takes rankings into account

Average reciprocal hit rank

- More user focused because users focus more on beginning of list instead of scrolling
- Penalize good recommendations that appear too low in the top N
 - Because user has to scroll to find them

Cumulative hit rate

- Throw away hits of predicted ratings if they are below a threshold
- We shouldn't get credit for recommending items a user won't enjoy
- Confined to ratings above a threshold

Rating hit rate

- Break down hit rate by predicted rating score

Other predictive factors

Recommendation-centric features

- Correctness
- Coverage
- Diversity
- Recommender confidence

User-centric

- Trustworthiness
- Novelty
- Serendipity
- Utility
- Risk

System centric

- Robustness
- Learning rate
- Scalability
- Stability
- Privacy

Delivery centric

- Usability
- User preference

Coverage

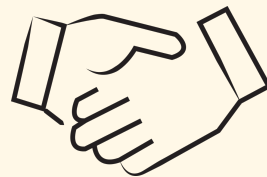
- Percentage of possible recommendations that the system can predict
- Example: a new movie can't enter a recommendation list until someone watches it and patterns are generated

Diversity

- Variety of items recommended
- High diversity = completely random

Novelty

- **How popular is the item recommended**
- **Must balance with trust**
 - Users want to see what they are familiar with to believe good recommendations



Churn



- How often do recommendations change?
- Showing the same recommendation all the time is a bad idea
- Must find a balance

Churn

- **If a user rates a new movie, does it change their recommendation?**
 - If yes, churn score is high




Responsiveness

- How quickly does new user behavior influence recommendations?
- Instant responsiveness is complex, hard to maintain and expensive

A metric is not an island

- Look at metrics together
- Understand trade-offs between them



Evaluate Recommender Systems ✓