Recommendation Systems Comparisons

Goals of Recommendation Systems

- Prediction Version: Predict values a user will give to an unrated item.
- Ranking Version: Determine top-k items for a particular user.

Types of User Data Used In Recommendation Systems

1- **Explicit Rating:** Rating specified by the user about their level of preference. Like and dislike for the item is explicit. The task is to predict the rating for the unrated items in the matrix.

2- **Implicit Rating:** Tracking of user behavior. A missing value does not mean a dislike for an item. As such it is recommended that the matrix is converted to a unary matrix (1,0) and missing values are filled with zeros to prevent overfitting.

Considerations

- ✓ Relevance
- ✓ Novelty
- ✓ Serendipity
- ✓ Diversity

Challenges

1- Sparsity
2- Cold Start
3- Fraud

Evaluation

1- MSE
2- RMSE
3- Precision
4- Recall
5- AUC
6- Kendall’s Tau (for rakings)

Recommender Type	Description	Approaches	Advantages	Disadvantages
Content-Based	It recommends items that are similar in content to those the user has liked in the past or matches their 'profile'.	• Naïve Bayes classifier • K-nearest neighbor • Decision Trees • Neural networks	• Easy to explain. • Provides personalized recommendations.	• Leads to recommendations that are 'too' similar. • Does dot have serendipity effects.
Collaborative Filtering	An user is recommended items based on past ratings of a group of users.	**User-Based:** the ratings provided by like-minded users of target A are used to make recommendations for A (i.e neighborhood based)	• Easy to implement and explain. • Exploits the logic of 'word of mouth' and user similarities.	• Do not scale well. • Computationally complex. • Does not work well in sparse matrixes.
		Item-Based: First determine the set of items S that are most similar to target item B. The ratings of set S items are used to determine whether an user will like target item B.	• Easy to implement and explain. • Recommendations make sense.	• Does not work well in sparse matrixes.
		Model Based: Decision trees, rule based, Bayesian & Latent Factor. They work within an optimization framework.	• Tend to work reasonably well with sparse data. • Works within optimization framework. • Latent models assume similarity between users is induced by a 'unknown' factor in the data.	• Need to regularize. • They are parametric models. • Can get stuck on local optimum, hard to know you reached global optimum.

Hybrid	Combine several approaches in providing recommendations.	Usually is done by combining content-based and collaborative filtering methods through weighting, cascading or zipping.	• Exploit strengths of several approaches and minimize weaknesses of other approaches.	• Complex to build and explain.
Knowledge Based	Recommendations based on explicit user requirements.			
Demographic Based	Recommendations based on user demographic attributes.			
Location Based	Take location into account when making recommendations.			
Time Sensitive	Take time into account when making recommendations.			
Group Based	Recommendations to a group (i.e find least miserable options for the group).			