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:** Ratings are specified by the user in regards 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- ROC
7- 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	A 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. • Highly engaged users can have a large impact. • 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 a user will like target item B.	• Easy to implement and explain. • Recommendations make sense.	• Computationally complex. • Does not work well for item sparse matrixes.
		Model Based: Decision trees, rule based, Bayesian & Latent Factor (matrix factorization, SVM, SVD). 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 cannot know if it has 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.		• Works well when the user is knowledgeable about the item domain. • Works well for high-price ticket items.	• Need a lot of information about the item.
Demographic Based	Recommendations based on user demographic attributes.		• Uses demographic data.	• Can lead to assumptions about certain demographic groups which may not be accurate for an individual.
Location Based	Take location into account when making recommendations.		• Takes time into account when making recommendations potentially leading to higher relevance and/or accuracy.	• Need location data.
Time Sensitive	Take time into account when making recommendations.		• Takes time into account when making recommendations potentially leading to higher relevance and/or accuracy.	• Needs time data. • It is longitudinal in nature. Need to be aware of time-effects.
Group Based	Recommendations to a group (i.e. find least miserable options for the group).		• Makes recommendations for an entire group. • Attempts to find the most acceptable option for the group.	• Works well in specific domains.

Sources

1- Melville P., Sindhwani V. (2017) Recommender Systems. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA

2- Felfernig A., Jeran M., Ninaus G., Reinfrank F., Reiterer S., Stettinger M. (2014) Basic Approaches in Recommendation Systems. In: Robillard M., Maalej W., Walker R., Zimmermann T. (eds) Recommendation Systems in Software Engineering. Springer, Berlin, Heidelberg

3- Aggarwal, Charu C. Recommender Systems. Chaper 1 (pg 1-28) ISBN 978-3-319-29659-3

4- Stanford Info Lab. Chapter 9: Recommendation Systems. http://infolab.stanford.edu/~ullman/mmds/ch9.pdf

5- McFee, Brian, Lanckriet, Gret. LARGE-SCALE MUSIC SIMILARITY SEARCH WITH SPATIAL TREES. https://bmcfee.github.io/papers/ismir2011_sptree.pdf

6- McFee, Brian, Lanckriet, Gret, Ellis, Daniel, Bertin-Mahieux. The Million Somg Dataset Challenge. http://eceweb.ucsd.edu/~gert/papers/msdc.pdf

7- Ellis, Daniel, Thierry Bertin-Mahieux. The Million Somg Dataset Challenge. http://eceweb.ucsd.edu/~gert/papers/msdc.pdf

8- Asanov, Daniar. (2018). Algorithms and Methods in Recommender Systems. https://www.snet.tu-berlin.de/fileadmin/fg220/courses/SS11/snet-project/recommender-systems_asanov.pdf

9- Khusro, Shah & Ali, Zafar & Ullah, Irfan. (2016). Recommender Systems: Issues, Challenges, and Research Opportunities. 1179-1189. 10.1007/978-981-10-0557-2_112.

10- Deep Learning based Recommender System: A Survey and New Perspectives. Zhang Shuai, Yao Lina, Sun Aixin. https://arxiv.org/pdf/1707.07435.pdf

11- B. Akay, O. Kaynar and F. Demirkoparan, "Deep learning based recommender systems," *2017 International Conference on Computer Science and Engineering (UBMK)*, Antalya, 2017, pp. 645-648.
doi: 10.1109/UBMK.2017.8093489