Evaluation Metrics for Personalized Recommendation Systems

Shuhao Jiang*, Jinlin Song²

¹Information Engineering College, Tianjin University of Commerce, Tianjin, China
²College of Science, Tianjin University of Commerce, Tianjin, China
*Corresponding author’s e-mail: jiangshuhao@tjcu.edu.cn

Abstract: This paper summarizes and reviews the latest research progress on the evaluation metrics of the recommendation system, expounds from multiple perspectives such as accuracy, diversity, novelty, and conducts an in-depth analysis of their respective advantages and disadvantages and applicable environments, focuses on the three levels of recommended diversity and evaluation criteria, and makes predictions on the development direction of the evaluation metrics of the recommendation system.

1. Introduction
In the context of the era of big data, it has become more and more convenient for people to use the Internet to obtain and publish information. However, it is precisely because of this convenience that people are facing the problem of information overload while being convenient for themselves, and a personalized recommendation system came into being. According to the user's historical behavior data, they can discover the user's hobbies and current needs, and recommend information and services that may be of interest to the user through various computer technology and data mining technology.

With the widespread application of recommendation systems, how to evaluate the pros and cons of a recommendation system has become a problem that researchers must face. When evaluating recommendation algorithms, user satisfaction, website revenue, and computational complexity need to be considered at the same time. The following summarizes and analyzes the current main evaluation metrics.

2. Accuracy evaluation metrics
Recommendation accuracy is the most basic metrics for evaluating a recommendation algorithm, which measures the extent to which the recommendation algorithm can accurately predict the user's preference for the recommended product. Based on the research of the existing literature, the accuracy metrics are divided into five categories, namely, prediction rating accuracy, classification accuracy, ranking accuracy, prediction rating relevance, and half-life utility metrics.

2.1. Prediction rating accuracy
The prediction rating accuracy metrics calculates the error between the algorithm prediction rating and the user's actual rating, which is the most important offline evaluation metrics for the recommendation system. Mean Absolute Error (MAE) [1] is an metrics based on this to calculate the absolute error between the user's actual rating and the predicted rating, and its definition is shown in formula (1).
Among them, $r_{ui}$ is the actual rating of the user $u$ on the product $i$, $r_{ui}'$ is the predicted rating given by the recommendation algorithm, and $T(u)$ is the user's behavior list in the test set.

In addition, Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Normalized Mean Absolute Error (NMAE) \[2\] are all metrics similar to MAE, their definition is shown in formulas (2) to (4):

\[
MSE = \frac{\sum_{u,i \in T} (r_{ui} - r_{ui}')^2}{|T|} \quad (2)
\]

\[
RMSE = \sqrt{\frac{\sum_{u,i \in T} (r_{ui} - r_{ui}')^2}{|T|}} \quad (3)
\]

\[
NMAE = \frac{MAE}{r_{\text{max}} - r_{\text{min}}} \quad (4)
\]

Among them, $r_{\text{max}}$ and $r_{\text{min}}$ are the maximum and minimum values of the user rating interval.

MSE and RMSE increase the penalties for users whose products rating are inaccurate in the prediction, and penalize larger errors more severely, so the evaluation of the system is more demanding. The ranges of MAE, MSE, and RMSE are all $[0, +\infty)$. NMAE normalizes the MAE within the rating interval, so that the performance of the same recommendation algorithm can be compared on different data sets.

2.2. Classification accuracy

The classification accuracy metrics calculate the correct proportion of users in the recommendation algorithm to determine whether they like the product. At present, the most commonly used classification accuracy metrics are precision, recall and F1.

 Billsus et al. \[3\] first introduced the precision and recall to the recommendation system and evaluated them. The precision calculates the ratio of the products that the user likes to all the recommended products in the recommended list. The recommendation precision of user $u$ is shown in formula (5):

\[
P(L_u) = \frac{L_u \cap B_u}{L_u} \quad (5)
\]

The recall calculates the ratio of the products that users like in the recommended list to all the products that users like in the system. The recommended recall of user $u$ is shown in formula (6):

\[
R(L_u) = \frac{L_u \cap B_u}{B_u} \quad (6)
\]

Among them, $L_u$ is the recommendation list of user $u$, and $B_u$ is the real data set of user $u$ in the test set.

Precision and recall are difficult to improve simultaneously. The two are often negatively correlated and depend on the length of the recommended list. Therefore, the F1 \[4\] is generally used to calculate the harmonic average of the two. The larger the value of F1, the better the prediction effect. The calculation is shown in formula (7):

\[
F_1 = \frac{2 \cdot PR}{P + R} \quad (7)
\]

Among them, $P$ is the precision and $R$ is the recall.

2.3. Sorting accuracy

The ranking accuracy metrics measure the degree of uniformity between the ordered recommendation list obtained by the recommendation algorithm and the user's ranking of products, and are suitable for evaluation of recommendation systems that need to provide users with a ranked list.
Zhou Tao et al. [5] proposed the average ranking rating to calculate the ranking accuracy of the recommender system. The ranking of user \( u \) to product \( i \) is shown in formula (8):

\[
ARS = \frac{L_i}{N}
\]  

(8)

Among them, \( N \) is the number of products not selected by the user in the training set, and \( L_i \) is the position of the product \( i \) to be predicted in the test set in the recommended list. The smaller the ranking rating, the more the system tends to rank the products that users like in the front.

Mean Average Precision (MAP) [6] means the search results returned, as shown in formula (9).

\[
MAP = \frac{\sum_{u \in U} AP_u}{|U|}
\]

(9)

\[
AP_u = \frac{1}{|\Omega_u|} \sum_{i \in \Omega_u} \frac{\sum_{j \in \Omega_u, b(p_{uj} < p_{ui}) + 1}{p_{ui}}}{6}\]

(10)

Among them, \( \Omega_u \) represents the search result, \( p_{ui} \) represents the ranking position of product \( i \) in the recommendation list, \( p_{ui} > p_{uj} \) represents that product \( j \) is ranked before product \( i \) in the recommendation list of user \( u \); \( U \) is user set. This metrics is sensitive to the position of the product. The higher the position of the product related to the user's preference in the list, the larger the value, which also indicates that the aggregate ranking performance of the algorithm is better.

2.4. Predictive rating relevance

Predictive rating correlation measures the correlation between the predicted rating and the user’s true rating. The three most common correlation metrics are Pearson correlation, Spearman correlation, and Kendall’s Tau. The Pearson correlation measures the degree of linear correlation between the predicted rating and the true rating, as shown in formula (11).

\[
PCC = \frac{\sum (r_i - \bar{r})(r'_i - \bar{r}')}{\sqrt{\sum (r_i - \bar{r})^2 \sum (r'_i - \bar{r}')^2}}
\]

(11)

Among them, \( r_i \) and \( r'_i \) respectively represent the real rating and predicted rating of product \( i \). The Spearman is defined in the same form as the Pearson, except that Spearman doesn’t consider the predicted rating value, but the ranking value obtained based on the predicted rating value, that is, replace \( r_i \) and \( r'_i \) in the above formula with the product \( i \) respectively real ranking and predicted ranking.

Similar to Spearman, Kendall’s Tau is another method of calculating ranking correlation. The larger the Tau, the better the system's prediction results. Formula (12) is an approximate method for calculating Tau [7].

\[
\tau = \frac{C-D}{\sqrt{(C+D+TR)(C+D+TP)}}
\]

(12)

Among them, \( C \) is the correct preference partial ordinal number predicted by the system, \( D \) is the preference partial ordinal number predicted incorrectly, \( TR \) is the number of products that the user has scored the same, and \( TP \) is the number of products with the same predicted value.

Although the calculation of predictive rating relevance is simple, there is still a problem of weak ranking. The so-called weak ranking refers to the situation where a user has the same rating for two or more products. In order to compare two different weakly ranking sequences, Yao YY [8] first proposed the Normalized Distance-based Performance Measure (NDPM). Its core idea is to compare the preference relationship between the recommendation system’s predicted ranking and the user’s actual ranking. The metrics based on preference relations are standardized, and the definition is shown
in formula (13).

\[ NDPM = \frac{2c^- + c^u}{2c^l} \] (13)

Among them, \( c^- \) is the number of product pairs that conflict between the system ranking and the user's ranking, \( c^u \) is the number of product pairs that are compatible in the two rankings, and \( c^l \) is the product pair that has a strict preference relationship in the user's actual ranking.

2.5. Half-life utility metrics

The half-life utility \([9]\) is proposed under the assumption that the probability of a user browsing a product and the specific ranking value of the product in the recommendation list are exponentially decreasing, and it measures the difference between the user’s real rating and the system’s default rating. The expected utility of user \( u \) is defined as shown in formula (14).

\[ HL_u = \sum \max \left( \frac{r_{ui} - d, 0}{2^{(l_{ui} - 1)/(h - 1)}} \right) \] (14)

Among them, \( r_{ui} \) represents the actual rating of user \( u \) for product \( i \), \( l_{ui} \) is the ranking of product \( i \) in the recommendation list of user \( u \), \( d \) is the default score, \( h \) is the half-life of the system. At present, there is no uniform standard for the use of half-life utility metrics.

In summary, prediction rating accuracy is suitable for systems that are more concerned with accurate prediction rating; classification accuracy is only suitable for systems with clear dichotomous preferences; ranking accuracy is suitable for systems that require strict recommendation ranking; Predictive rating relevance is suitable for systems that don’t pay attention to accurate predictive rating, where NDPM is suitable for weak ranking; the half-life utility metrics is that if all results of the recommendation list are displayed in a text list of several pages, then the website designer needs to simulate the way users scan the list to calculate the half-life utility score.

3. Diversity evaluation metrics

In order to satisfy users' extensive interests, the recommendation list needs to be able to cover users' different interest areas, that is, the recommendation needs to be as diverse as possible \([10]\). The diversity in the recommendation system includes three levels: individual diversity, aggregate diversity and temporal diversity \([11]\).

3.1. Individual diversity

Individual diversity measures the diversity of recommendation results from the perspective of a single user, focuses on whether products in different fields can be recommended to users, aims to avoid recommending products that are too similar to individual users, improves user experience, and increases user satisfaction.

Assuming that \( s(i, j) \in [0, 1] \) defines the similarity between products \( i \) and \( j \), then the diversity of recommendation list \( L_u \) of user \( u \) is shown in formula (15).

\[ Diversity(L_u) = 1 - \frac{\sum_{l \in l_u} \max_{i \neq j} s(i, j)}{2^{|l_u|(|l_u| - 1)}} \] (15)

Diversity describes the dissimilarities between the two products in the recommended list. Therefore, diversity and similarity correspond.

3.2. Aggregate diversity

The aggregate diversity is based on the aggregate consideration of the recommendation results, calculates the total number of products in the recommendation list of all users as a proportion of all products, and mainly emphasizes that the recommendations for different users should be as different as possible. The improvement of the aggregate diversity can avoid the recommendation results from being concentrated on a few popular products, which is conducive to the discovery of unpopular products by users.
3.2.1. Coverage
Coverage\cite{12} describes the ability of a recommendation system to discover the long tail of products. The simplest definition of coverage is the proportion of products recommended by the recommendation system to the total product collection.

Assuming that the user set of the system is $U$, the recommendation system recommends a list $L_u$ of length $n$ to each user, then the coverage of the recommendation system is shown in formula (16).

$$\text{Coverage} = \frac{|\cup_{u \in U} L_u|}{n}$$

(16)

Coverage is a metric that content providers will care about. A recommendation system with a coverage rate of 100% can recommend every product to at least one user.

3.2.2. Hamming Distance
The aggregate diversity measures the ability of the recommendation system to recommend different products to different users, and the individual diversity measures the variety of products recommended by the recommendation system to a single user. Hamming distance\cite{7} can measure the different degrees of two user recommendation lists, and its definition is shown in formula (17).

$$H_{uv} = 1 - \frac{Q_{uv}}{|L|}$$

(17)

Among them, $L$ is the length of the recommendation list; $Q_{uv}$ is the number of the same products in the two recommendation lists recommended by the system to users $u$ and $v$. The diversity of recommendation list is the average value of $H_{uv}$. The larger the value, the higher the aggregate diversity.

3.3. Temporal Diversity
The user’s interest in the product is constantly changing over time, and there may be a large gap between past preferences and current preferences. Therefore, it is unscientific to treat all behavioral data of target users equally. Users always hope to obtain new recommendation results that are different from the past. In recent years, the time perception in the recommendation system has become a research hotspot, and the effective improvement of the temporal diversity of personalized recommendation\cite{10} has also received increasing attention. Its definition is shown in formula (18).

$$\text{SSD}(L|u) = \frac{|L_t \cap L_{t-1}|}{|L_t|}$$

(18)

SSD refers to the proportion of recommended products that aren’t included in the previous recommendation list, and mainly examines the temporal diversity of the recommended results. Among them, $L_{t-1}$ is the last recommendation of $L_t$, $L_t \cap L_{t-1} = \{r \in L_t | r \notin L_{t-1}\}$. The smaller the SSD value, the better the temporal diversity of the recommendation list.

The increase of individual diversity will increase user satisfaction, make recommendations novel and not tedious, increase user stickiness, and ultimately increase website revenue; the increase in aggregate diversity will increase the proportion of items in the recommendation list and the exposure of website platform products increases, especially for long-tail products. The time factor must be considered when a website recommends items for users. The user's interest is rarely static. Adding a time factor when recommending helps improve the accuracy of the recommendation.

4. Other Evaluation Metrics
In addition to recommendation accuracy and diversity, recommendation system evaluation has some other important metrics, including recommendation novelty, surprise degree, and so on. This paper refers to these metrics as other evaluation metrics.

4.1. Novelty
Novelty recommendation refers to recommending products that they have not heard of before. Celma Òscar et al.\cite{13} studied the evaluation of novelty. The easiest way to evaluate novelty is to calculate the average popularity of the products in the recommended list, because the less popular products are more likely to make users feel novel.
Among them, $p(i)$ is the popularity of product $i$, and $L_u$ is the recommendation list of user $u$. 

4.2. Surprise degree

Surprise degree is the most popular topic in the field of recommendation systems in recent years. If the recommendation result isn’t similar to the user's historical interests, but the user feels satisfied, it can be said that the recommendation result has a high surprise degree. The surprise degree brought by product $i$ to user $u$ is shown in formula (20):

$$S_{u,i} = \frac{\sum_{i=1}^{U} R_{u,i} - \min_{i,j} s(i,j)}{\sum_{i=1}^{U} R_{u,i} - \max_{i,j} s(i,j)}$$  \hspace{1cm} (20)

Among them, $R_{u,i}$ represents the rating of user $u$ on product $i$, $U$ represents the number of users who have rated product $i$, $Q$ represents the number of ratings that user $u$ has given, $R_{u\text{max}}$ represents the highest rating that user $u$ has given, $s(i,j)$ is the similarity of the two commodities $i$ and $j$.

4.3. Trust

Trust is the user’s sense of recognition of the recommendation system’s results. For automatic recommendation systems based on machine learning, there is a problem of trust. If users trust the recommendation system, it will increase the interaction between the user and the recommendation system. Generally speaking, in social networks, if a user is trusted by more other users, it generally indicates that the credibility of this user is higher. The user’s trust level is shown in formula (21) \cite{14}:

$$T_{u,v} = \frac{1}{\sum_{u \in T_u} |T_u|} |T_u|$$ \hspace{1cm} (21)

Among them, $T_u$ represents the set of trusted users of user $u$; $|T_u|$ represents the size of the set; $N_u$ and $N_v$ represent the number of trusted users of users $u$ and $v$, respectively.

5. Conclusions

This paper systematically introduces the evaluation metrics of the recommendation system, and summarizes it from multiple perspectives such as accuracy and diversity. So far, different recommendation algorithms still have no unified metrics to compare the advantages and disadvantages. In the test, scholars often choose different evaluation metrics according to the different task requirements of the recommendation system. But whether it is to improve the accuracy, diversity or other of the recommendation algorithm, the ultimate goal is to improve user satisfaction. After all, business benefits increase with the increase in user satisfaction.

References

[1] Herlocker, J. L., Konstan, J. A., Borchers, A., Riedl, J. (2017). An algorithmic framework for performing collaborative filtering. In: ACM SIGIR Forum. New York. pp. 227-234.

[2] Balabanović, M., Shoham, Y. (1997) Fab: content-based, collaborative recommendation. Communications of the ACM, 40(3): 66-72.

[3] Billsus, D., Pazzani, M. J. (1998). Learning collaborative information filters. In: Icml. Madison. pp. 46-54.

[4] Hanley, J. A., McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, 143(1), 29-36.

[5] Zhou, T., Jiang, L. L., Su, R. Q., Zhang, Y. C. (2008). Effect of initial configuration on network-based recommendation. EPL (Europhysics Letters), 81(5): 58004.

[6] Yu, D., Cheng, T., Yuan, X. (2020) Software Crowdsourcing Task Recommendation Algorithm Based on Learning to Rank. Computer Science, 47(12): 106-113.

[7] Liu, J., Zhou, T., Guo, Q., Wang, B. (2009) Overview of the Evaluated Algorithms for the Personal Recommendation Systems. Complex Systems and Complexity Science, 6(03): 1-10.
[8] Yao, Y. Y. (1995) Measuring retrieval effectiveness based on user preference of documents. Journal of the American Society for Information Science, 46(2): 133-145.

[9] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. New York, pp. 285-295.

[10] Lin, H., Chen, L., Lan, X., Zhang, J., Yin, H. (2017) An optimization collaborative filtering algorithm based on user context and temporal diversity. Journal of Northwest University (Natural Science Edition), 47(06): 822-828.

[11] An, W., Liu, Q., Zhang, L. (2013) Review on Diversity in Personalized Recommender Systems. Library and Information Service, 57(20): 127-135.

[12] Ge, M., Delgado-Battenfeld, C., Jannach, D. (2010). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: Proceedings of the fourth ACM conference on Recommender systems. New York. pp. 257-260.

[13] Celma, O. (2010). Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space. Springer, New York.

[14] Yuan, N., Xin, L., Wang, H., Ning, P. (2020) Collaborative filtering recommendation algorithm based on user interestingness and believability. Computer Engineering and Design, 41(07): 1967-1974.