Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230861

Overview of definition, evaluation, and algorithms of serendipity in recommender systems

Ning Sun

University of Alberta, Edmonton, Canada, T6G 2R3

ning2@ualberta.ca

Abstract. Over time, recommendation systems are playing an important role in an increasingly wide range of areas, such as paper retrieval sites that can recommend papers or books to users, and shopping sites that can recommend products to users. With the development of recommendation systems, there are many different metrics to measure a good recommendation system, including serendipity. This paper summarizes the definition of serendipity, a review of the metrics for measuring serendipity, and several major serendipity-oriented algorithms and presents conjectures for future research on serendipity. Through the research of some papers, for how to delimit and evaluate recommender systems, experts have mostly focused on the unexpected, and most of them use and optimize collaborative filtering algorithms to achieve and improve serendipity.

Keywords: recommender system, serendipity, systematic literature review, content-based filtering, collaborative filtering, greedy algorithm.

1. Introduction

A good recommendation system has many indicators, one of which is relatively important is user satisfaction and accuracy rate. However, if the system focuses too much on the accuracy of recommendations, or recommends a large number of popular items, it will make users bored [1]. According to Kotkov's definition, when a user is frequently recommended items that match his interests or personal information or when always recommended popular items, they are less satisfied and get bored, which is the problem of overspecialization, which can reduce user satisfaction and can result in the loss of business value of the recommender system [1]. Ziarani et al. argue that argue that novelty and diversity are two ways to solve overspecialization. However, to make users more satisfied, scientists started to consider recommending serendipitous items, and serendipity became one of the standard to evaluate recommended systems [2]. As the recommendation system becomes more and more mature, serendipity gradually attracts people's attention, because which effectively increases people's satisfaction with the recommendation system, and this paper asks the following questions: 1. what is the definition of serendipity; 2. how to evaluate serendipity; 3. how to implement serendipity, and what are the popular algorithms. This paper will review scientists' definition of recommender systems, especially serendipity; 3. Review the criteria for evaluating serendipity; 4. Review the algorithms for serendipity.

DOI: 10.54254/2755-2721/6/20230861

2. Definition of serendipity

The importance of serendipity was first demonstrated and summarized by Toms in 2000 [3]. In addition, later scientists gradually refined the definition of serendipity. KAMINSKAS and BRIDGE summarized the literature over the years and came up with 3 elements: 1. The classifier does not determine the relevance of the item to the user's preferences; 2. Whether the item covers all areas of interest to the user; 3. Whether the item has the intersection of the features of the two input items [4].

Yaqub classifies serendipity into: Walpolian, Mertonian, Bushian, and Stepha. Walpolian is something that the explorer finds accidentally. Mertonian is the explorer who solves the problem by an accidental method. Bushian is similar to Mertonian, but Bushian is the accidental discovery of the answer [5]. The accidental invention of saccharin mentioned by the author in the paper is Bushian Stepha, on the other hand, discovered solutions and problems by accident, such as what Yaqub mentioned in his paper, in which the inventor first discovered non-shattering flask, and then learned about the seriousness of the injuries caused by flying glass, so he subsequently invented safety glass.

According to Kotkov's summary, contingency items have the following three keywords for users: relevant, novel, and unexpected [6]. According to his summary, an item is serendipity to the user if the following three conditions are met: 1. The user has never touched the item; 2. The user has access to the item but has never consumed or learned about it; 3. The user does not have the impression that he has consumed or learned about the item. Some controversy about the definition of serendipity. The biggest point of contention is relevance. Some scientists believe that the definition of serendipity is novelty and unexpected , but most scientists believe that relevance and unexpected is serendipity, and some scientists believe that serendipity has all three characteristics at the same time [1] [2][7].

3. Serendipity assessment methods and criteria

Serendipity can be assessed in many ways. First of all, the most direct method is user study, which can get the most subjective and direct feelings of users by conducting a documented survey, and it is less risky. The following conclusions were drawn: 1. All unexpectedness and novelty expand users' preferences and make them like other kinds of items. In Kotkov's prediction, he found that accidental films are more liked than non-accidental films, which indicates to some extent that users are still interested in serendipitous items [6]. However, user study also has some disadvantages. For example, it is more expensive to recruit test users and difficult to organize users to be surveyed on a large scale. Moreover, double-blind experiments are difficult to implement and reflect the real environment. The second method is the calculation by formula. The most common formula is:

$$\frac{|R_{unexp}| \cap |R_{useful}|}{|R|} \tag{1}$$

But scientists have their own criteria, and Kotkov summarizes the measure of unexpectedness:

$$PMI(i,j) = -log_2 \frac{p(i,j)}{p(i)p(j)} / log_2 p(i,j) \tag{2}$$

where P(i) is the likelihood that the user will rate item i, what is the probability that the user will rate item i. p(i, j) is the likelihood that the user will rate item i and item j together. PMI(i, j) ranges from 1 to -1. When PMI is 1, it means that the user always rates i and j together, and when PMI is -1, it means that the user never rates i and j together. This formula calculates the level of similarity between two items (i and j). If the result tends to -1, the less similar the two items are, and if it tends to 1, the more similar the two items are. Two variants are extended [8]:

$$unexp_{kam}^{co-occ1}(i,u) = maxPMI(i,j), j \in I_u \tag{3}$$

$$unexp_{kam}^{co-occ2}(i,u) = \frac{1}{|I_u|} \sum_{j \in I_u} PMI(i,j) \tag{4}$$

Kotkov also summarized the complete criteria by examining Murakami's serendipity metric algorithm:

$$Ser_{mur}(u) = \sum_{i \in R_u} \max(Pr_u(i) - Prim_u(i), 0) \cdot rel_u(i) \tag{5}$$

Where Ru is the list of recommendations; Pr_u is the confidence primitive models that recommend items to users by examined models. Prim_u is the confidence primitive models that recommend items to users by primitive models. rel _u (i) is 1 if item i is related to user u, 0 otherwise. Later, through modifications by Ge and Adamopoulos et al., this metric formula became:

$$ser_{ad}(u) = \frac{1}{|R_u|} \sum_{i \in (R_u \setminus (E_u \cup PM))} rel_u(i) \quad (6)$$

Where i is item. PM is the set of items recommended by the initial recommender system model, and Eu is the item i that belongs to the interests of user u. If item i is related to user u, it is 1, if not, it is 0. Chantanurak et al. They evaluate serendipity is assessed by counting the ratio of the number of useful recommendations of the recommender system to the number of recommended items that users are surprised by [2]:

$$Ser@N = \frac{\#useful}{\#unexpected} \quad (7)$$

where #useful is the number of items for which the user expressed a preference or interest in the recommended item. The meaning of #unexpected is the number of items that the user was surprised by the recommended item.

Niu and Abbas proposed a framework to calculate serendipity scores:

$$surprise(d) = 1 - familiarity(d) * expectation(d) \quad (8)$$

Where d is one of the items recommended by the whole recommendation system. The core of this framework is Curiosity Model, and the purpose of this model is to find the parameters. θ and δ, where θ represents the number of curiosity parameters and the latter represents the type, so as to find the maximum value of curiosity:

$$(\theta', \delta') = \arg\max(curiosity(surprise(\theta, \delta)) \quad (9)$$

After obtaining the curiosity model, they obtained the following equation:

$$serendipity(d) = value(d) - abs(surprise(d) - surprise(\theta', \delta')) \quad (10)$$

Where d is the item with the highest serendipity score [9].

4. Overview of serendipity methods and algorithm integration

The algorithms of recommendation systems have been developed for many years, and one of the most common and classic algorithms should be the collaborative filtering algorithm. The basic principle is based on the data of user's historical behavior.

4.1. User-based collaborative filtering algorithm

Collaborative filtering algorithms mainly include two types of algorithms: user-based collaborative filtering algorithm and item-based collaborative filtering algorithm. Among them, the user-based collaborative filtering algorithm is to recommend items to users with similar interests. This algorithm is not suitable for improving the serendipity metric of a recommender system, because it is difficult for two users with the same interests to find items with Serendipity, since they may be familiar with both in this area. But Afridi studied how to make users control the serendipity of the recommendation system, and it used the recommendation slider to allow users to achieve control over the desired serendipity and accuracy of the recommendation system. Users can control the item list to reorder and then generate a list of unexpected recommendations. In this process, Afridi uses a collaborative filtering algorithm to provide recommendations to the user. He performed MANOVA test on user-accepted data and got the result that enhanced user control helps to improve the problem of overspecialization of recommendation systems and can enhance the chance [10].

4.2. Item-based collaborative filtering algorithm

The basic idea of item-based collaborative filtering algorithm is to pre-calculate the similarity between items based on the historical preference data of all users, and then recommend items similar to the user's favorite item to the user. The similarity is generally calculated by the cosine similarity method. Matrix Factorization, which means decomposing the matrix into the product of several matrices, can decompose the collaborative filtering co-occurrence matrix into more matrices to discover more invisible features. According to the experiments done by Kotkov, they chose a music-related domain to collect two datasets, the Vkontakte dataset and the Last.fm dataset. First, using ItemCF, the items were sorted by similarity to the items selected by users. Each recording (item) is represented as a vector in a multi-dimensional feature space, and FEATURE is the user's rating of the item. He refers to Vkontakte's dataset as VK, so the recording of VK is represented as:

$$i^{vk} = (u_{1,i}^{vk}, u_{2,i}^{vk}, \ldots, u_{n,i}^{vk}) \tag{11}$$

The elements in each recording are denoted as:

$$u_{k,i}^{vk} \in \{0,1\}\ for\ k = 1, \ldots, ||U|| \tag{12}$$

Where k is the user and U is the set of users, which means that each element has two values, 0 or 1. So, if Vk user k selects i^vk, the element u_(k,i)^vk in it becomes 1, or 0 if it is not selected and they also use a dataset Last.fm, which becomes FM, and merge the users of FM into vk's recording, Kotkovde the two datasets would then become:

$$i^{vkfm} = (u_{1,i}^{vk}, u_{2,i}^{vk}, \ldots, u_{n,i}^{vk}, u_{1,i}^{fm}, u_{2,i}^{fm}, \ldots, u_{n,i}^{fm}) \tag{13}$$

When the user has finished selecting, the result of the user's selection can be clearly observed by 1 and 0. Then, the collaborative filtering algorithm sorts the other recordings according to the one selected by users, and the rule of sorting is the similarity of the other recordings with the one selected by users. Kotkov et al. chose conditional probability as a similarity measure, and the formula is as follows:

$$p(i,j) = \frac{Freq(i \wedge j)}{Freq(i) \cdot Freq(j)^{\alpha}} \tag{14}$$

where Freq(i) represents how many users have selected item i. Similarly, Freq(j) represents how many users have selected item j. Freq(i∧j) represents how many users select item i and item j at the same time. The role of α is to reduce the similarity of popular items. Kotkov chose α to be 1 in his experiments. In this experiment, there are more FM users than Vk users. Kotkov devised this rule to compare therecording and derive the value of similarity:

$$sim(i,j) = \begin{cases} p(i^{vk}, j^{vk}), \exists i^{vk} \wedge \exists j^{vk} \wedge (\nexists i^{fm} \vee \nexists j^{fm}) \\ p(i^{fm}, j^{fm}), \exists i^{fm} \wedge \exists j^{fm} \wedge (\nexists i^{vk} \vee \nexists j^{vk}) \\ p(i^{vkfm}, j^{vkfm}),\ \exists i^{vk} \wedge \exists j^{vk} \wedge \exists i^{fm} \wedge \exists j^{fm} \end{cases} \tag{15}$$

Then, the sum of similarity was obtained:

$$score(u,i) = \sum_{j \in I_u} sim(i,j) \tag{16}$$

Where I_u is the set of items selected by user u. Kotkov's algorithm uses this score to sort [11].

4.3. Content-based recommender systems

Sutter et al. Content-based recommender systems were outlined in the paper, where content-based recommender systems use two sets of metadata, one from a set of users Ua, and a set of data from a set of items I (represented as keywords) to extract current data. There are items to make predictions about the target user's items [12].

Term Frequency Inverse Document Frequency (TF-IDF) is the most commonly used term weighting scheme, Lops et al. explain the following in their paper:

DOI: 10.54254/2755-2721/6/20230861

• Rare words are not less relevant than common words (IDF assumption);

• Multiple occurrences of a word in a document are not less relevant than single occurrences (TF assumption);

• Long documents are preferred over short documents (normalization assumption) (12).

Kotkov also did a Content-Based Filtering method in the same article. This method is based on item attributes to make recommendations, such as tags of items, etc. In Kotkov's experiments, this attribute is VK - FM artists and FM tags. First, frequency inverse document frequency (TF-IDF) is used. There are the following two components:

$$tfidf_{attr,i} = tf_{attr,i} \cdot idf_{attr} \tag{17}$$

Where attr is attribute; Where $tf_{attr,i}$ is the frequency of the attribute of item i; idf_{attr} is the inverse frequency of the attribute, also called inverse document frequency.

In Kotkov's experiment, each FM artist corresponds to a particular VK artist:

$$tf_{zttr,i} = \frac{n_{attr,i}}{n_i} \tag{18}$$

where ni represents the number of attributes of item i, and nattr,i represents the proportion of this attribute in the attributes of item i. In Kotkov's example, nattr,i = 1 for each item, and ni varies from item to item. The inverse document frequency and the number of items with attributes in the dataset are related.

$$idf_{attr} = ln\frac{||I||}{||I_{attr}||} \tag{19}$$

Where I is the set of all items; Iattr is the set of items with attributes, and inverse document frequency is chosen. The reason is the difference between rare attributes and popular attributes. Then, Kotkov integrated the FM tags in order to consolidate the data into:

$$i^{at} = (a_{1,i}, a_{2,i}, \dots, a_{d,i}, t_{1,i}, t_{2,i}, \dots, t_{q,i}) \tag{20}$$

where $t_{k,i}$ is TF IDF weight;

The user vector is:

$$u^{at} = (a_{1,u}, a_{2,u}, \dots, a_{d,u}, t_{1,u}, t_{2,u}, \dots, t_{q,u}) \tag{21}$$

Where $t_{k,u}$ is how many recordings with tag tk are selected by the user.

Kotkov uses cosine similarity to compare similarity:

$$\cos(u, i) = \frac{u \cdot i}{||u|| \cdot ||i||} \tag{22}$$

Where u is the user vector and i is the item vector.

Kotkov's experimental results show that for collaborative and content-based filtering algorithms, serendipity increases only when items in source and target domains overlap, and the more items overlap, the higher the accuracy of the algorithm [13].

Also, kotkov describes the chance-oriented greedy algorithm. This algorithm describes an accuracy-oriented algorithm rating first into a list that generates the list $RSu(n)$ with accuracy as a criterion, and then iterates with SOG from $RS\ uu(n)$ to select items to generate a list Res with diversity as a criterion. The SOG generates a candidate set during the iteration which contains the items in Res and the highest scoring candidates. So the order of Res will be different from $RSu(n)$ for the purpose of serendipity.

Kotkov also proposes to define the chance in terms of score and defines the parameters of it as follows:

$$score_{uiB} = a_{rel} \cdot \hat{r}_{ui} + \alpha_{div} \cdot div_{iB} + \alpha_{prof} \cdot prof_{ui} + a_{unpop} \cdot unpop_i \tag{23}$$

Where a is a parameter; the table below represents the different parameters; rel represents relevance; div represents diversity; prof represents how dissimilar items are from user profiles, and unpop represents unpopular parameters.

$\hat{\boldsymbol{r}}_{\boldsymbol{ui}}$ is the only evaluation of item i by user u, $\boldsymbol{div_{iB}}$ represents the difference between item i and dataset B, $\boldsymbol{prof_{ui}}$ i and user's usual consumption, and $\boldsymbol{unpop_i}$ represents the degree of unpopularity of item i:

$$div_{iB} = \frac{1}{||B||}\sum_{j\in B} 1 - sim_{i,j} \tag{24}$$

Where $\boldsymbol{sim_{i,j}}$ is a similarity measure in the range of [0,1];

$\boldsymbol{prof_{ui}}$ and $\boldsymbol{unpop_i}$ are respectively calculated by the following equations:

$$\boldsymbol{prof_{ui}} = \frac{\boldsymbol{1}}{||\boldsymbol{I_u}||}\sum_{j\in \boldsymbol{I_u}} \boldsymbol{1} - \boldsymbol{sim_{i,j}} \tag{25}$$

$$\boldsymbol{unpop_i} = \frac{||\boldsymbol{U_i}||}{||\boldsymbol{U}||} \tag{26}$$

U is a set of users for a specific recommender system at a specific time [1].

5. Discussion

This paper summarizes the indicator serendipity of the recommender system and demonstrates that scientists have made rapid progress in serendipity research results this year. However, there seem to be a lot of points for development. The author personally think that the recommendation system serves the user, and the user's satisfaction is the fundamental purpose pursued by the recommendation system. So the paper thinks the future trend will be that the serendipity of recommendation systems will gradually be user-centered and adapted to different users for the recommendation. Secondly, in Kotov's article, the parameter is mentioned as personal information about the user, but he did not conduct a study on this aspect. While Nguyen et al. studied the demand for users' individual personality traits on recommendation systems, they surveyed 1800 users. They rated two groups of movie lists to users, one group containing 12 movies and one group containing 15 recently rated movies, using cosine similarity to calculate:

$$serendipity\ score = \frac{1}{12\times 15}\sum_{m_i\in S}\sum_{r_j\in R} cosine\ distance(m_i, r_j) \tag{27}$$

Nguyen concluded that most high-conscientious users prefer low-serendipity recommendations, while most low-conscientious users prefer high-serendipity recommendations [14]. If the results of this study can be combined with Kotkov's greedy algorithm, it may increase user satisfaction, which could greatly improve recommender systems.

6. Conclusion

This paper collects and finds the research results of scientists in recent years and draws the following conclusions. There are many definitions of serendipity, but most of them include the unexpected and relevance. The evaluation methods are influenced by the definitions, and scientists have their own criteria. But most of them include unexpected parameters. So, the unexpected is an important indicator of serendipity. For the algorithms of serendipity, scientists mostly use item-based collaborative filtering algorithms or content-based recommendation systems. Some scientists also try to improve the serendipity of recommendation systems by being user-centered. Firstly, the author should put the above ideas into experiments to verify the specific effects. Secondly, according to Nguyen's article, users with different personalities have different needs for serendipity. If users are not interested in the items recommended by the system by chance but often receive these kinds of recommendations, their satisfaction with the recommendation system will be greatly reduced. So in the future, people should also put the research into the combination of serendipity and user satisfaction.

References

[1] Kotkov, D., Veijalainen, J. & Wang, S. (2020). How does serendipity affect diversity in recommender systems? A serendipity-oriented greedy algorithm. Computing 102, 393–411.

[2] Ziarani, R. J., & Ravanmehr, R. (2021). Serendipity in recommender systems: a systematic literature review. Journal of Computer Science and Technology, 36(2), 375-396.
[3] Toms, E. G. (2000). Serendipitous Information Retrieval. DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries.
[4] Kaminskas, M., & Bridge, D. (2016). Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. ACM Transactions on Interactive Intelligent Systems (TiiS), 7(1), 1-42.
[5] Kotkov, D., Konstan, J. A., Zhao, Q., & Veijalainen, J. (2018, April). Investigating serendipity in recommender systems based on real user feedback. In Proceedings of the 33rd annual acm symposium on applied computing (pp. 1341-1350).
[6] Zheng, Q., Chan, C. K., & Ip, H. H. (2015, July). An unexpectedness-augmented utility model for making serendipitous recommendation. In Industrial conference on data mining(pp. 216-230). Springer, Cham.
[7] Kotkov, D., Wang, S., & Veijalainen, J. (2016). A survey of serendipity in recommender systems. Knowledge-Based Systems, 111, 180-192.
[8] Niu, X., & Abbas, F. (2017, July). A framework for computational serendipity. In Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization (pp. 360-363).
[9] Afridi, A. H. (2018). User control and serendipitous recommendations in learning environments. Procedia computer science, 130, 214-221.
[10] Lops, P., Gemmis, M. D., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. Recommender systems handbook, 73-105.
[11] Saat, Nur Izyan Yasmin, Shahrul Azman Mohd Noah and Masnizah Mohd. “Towards Serendipity for Content–Based Recommender Systems.” International Journal on Advanced Science, Engineering and Information Technology (2018): n. pag.
[12] Kotkov, D., Wang, S., & Veijalainen, J. (2016, April). Improving serendipity and accuracy in cross-domain recommender systems. In International Conference on Web Information Systems and Technologies (pp. 105-119). Springer, Cham.
[13] Nguyen, T. T., Maxwell Harper, F., Terveen, L., & Konstan, J. A. (2018). User personality and user satisfaction with recommender systems. Information Systems Frontiers, 20(6), 1173-1189.