Adaptive Serendipity for Recommender Systems: Let It Find You

Miriam El Khoury Badran[1], Jacques Bou Abdo[2], Wissam Jurdi[2] and Jacques Demerjian[3]
[1]*Department of Computer Science, Notre Dame University, Zook Mosbeh, Lebanon*
[2]*Department of Computer Science, Notre Dame University, Deir el Qamar, Lebanon*
[3]*LARIFA-EDST Laboratory, Faculty of Science, Lebanese University, Fanar, Lebanon*

Keywords: Serendipity, Accuracy, Recommender System.

Abstract: Recommender systems are nowadays widely implemented in order to predict the potential objects of interest for the user. With the wide world of the internet, these systems are necessary to limit the problem of information overload and make the user's internet surfing a more agreeable experience. However, a very accurate recommender system creates a problem of over-personalization where there is no place for adventure and unexpected discoveries: the user will be trapped in filter bubbles and echo rooms. Serendipity is a beneficial discovery that happens by accident. Used alone, serendipity can be easily confused with randomness; this takes us back to the original problem of information overload. Hypothetically, combining accurate and serendipitous recommendations will result in a higher user satisfaction. The aim of this paper is to prove the following concept: including some serendipity at the cost of profile accuracy will result in a higher user satisfaction and is, therefore, more favourable to implement. We will be testing a first measure implementation of serendipity on an offline dataset that lacks serendipity implementation. By varying the ratio of accuracy and serendipity in the recommendation list, we will reach the optimal number of serendipitous recommendations to be included in an accurate list.

1 INTRODUCTION

Nowadays, with the internet being used world widely and for many applications, the user is exposed to a very large quantity of information. Consumers are suffering from what is called information overload. The need to bridge the gap between the demand and the supply becomes of urging importance. Recommender systems arise in order to predict what the user might need the most and recommend it to him, narrowing consequently his choices. Personalization of the internet's content or information-filtering has a very important role in knowledge management (Reviglio, n.d.). The personalization happens in two ways: explicitly through the act of rating or implicitly through activity monitoring with the use of artificial intelligence and machine learning. Personalization is somewhat dangerous, especially when done implicitly since it is imposed on the user who might not desire it. It creates filter bubbles and echo rooms. In the filter bubbles, the user continues to see and listen to what reinforces his interest and opinion. While the echo room is a group situation where information, ideas, and beliefs are being amplified like the actual echoing phenomenon. If used up to a certain extent, personalization brings satisfaction to most users; however, if techniques continue to diverge towards further enhancing it, the result would be a dangerous over-personalized environment having users that are addicted to their comfort zone.(Reviglio, n.d.). Customers of e-retail businesses will view only their familiar items without being exposed to new items that they don't even know exist even though these new items may solve problems that customers face and they aren't aware that these problems are solvable. Serendipitous items will satisfy customer's needs and increase sales. That's why "beyond-accuracy" objectives are essential in recommender systems. Kaminskas and Bridge analyze these objectives: diversity, serendipity, novelty, and coverage (Kaminskas and Bridge, 2016).

Serendipity is commonly described as "pleasant surprise", "unintended finding", "accidental discovery" or simply the "Aha!" experience (Sun et al., n.d.). The term was first used in 1754 by Horace

Badran, M., Abdo, J., Jurdi, W. and Demerjian, J.
Adaptive Serendipity for Recommender Systems: Let It Find You.
DOI: 10.5220/0007409507390745
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 739-745
ISBN: 978-989-758-350-6

in his book The Three Princes of Serendipity, whose adventure was full of unexpected happy discoveries. Simply put, serendipity is knowing what the user doesn't know he/she likes: a hard task indeed.

The item inside the user's mind we be divided into two categories (for the sake of simplicity): what he/she knows and what he/she ignores. And each category can be divided to two sub-categories: what he/she likes or dislikes for the known items and what he/she would like or would dislike for the unknown. Serendipity lies in the subcategory of the items the user ignores but would like. It refers to the process of "finding valuable or pleasant things that are not looked for" as defined by Van Andel (Kaminskas and Bridge, 2016).

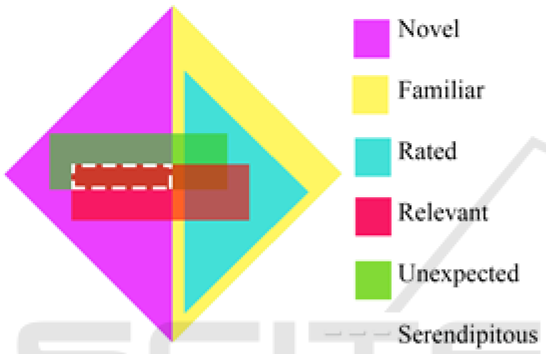

Figure 1: Recommended items from a user's point of view.

Serendipity is the intersection of what is unexpected and relevant at the same time as shown in Fig 1.

Users tend to enjoy what is relevant and accurate, unaware that there might be an entire new world that they might be interested in, but that they have never discovered.

For all the previously mentioned reasons, and considering the importance of serendipity in a world so accurate that it is becoming boring and redundant, we suggest integrating some serendipitous items in the recommendation list. The purpose of this paper is, first, to show that serendipity can increase user satisfaction even in offline datasets that aren't linked to serendipity studies. The second goal is to test the optimal number of unexpectedly relevant items among others that are accurate.

This paper is divided as follows: in section 2 we discuss the background and the related work. Then, we show the implementation environment including the algorithm in its steps, and the dataset. In the last section, the experimental results will be presented followed by the limitations.

2 BACKGROUND AND RELATED WORK

In the current section, we have an overview of the previous studies and works that are related to serendipity.

Serendipity is something hard to define and this complexity in the definition impacts the possibility of implementation. Ge et al. (Ge et al., 2010) indicate that experimental studies of serendipity are very rare since it is not only hard to define, but, in parallel, hard to measure. This difficulty to define and measure surprise and unexpectedness was mentioned in other surveys and studies (Kaminskas and Bridge, 2016). As previously mentioned, many research studies are trying to grasp the meaning of this happy surprise; they all admit that it is somewhere between the unexpectedness, the novelty and the relevance or what is also called utility or usefulness.

Kotkov et al. in their survey list state-of-the-art recommender approaches that suggest serendipitous items (Kotkov et al., 2016). They point at the re-ranking algorithm, opposite to the accuracy-based algorithms, where obvious suggestions are given a low ranking. This algorithm can use any accuracy algorithm to give the result, and in case we desire a serendipity-oriented modification, specific algorithms are to be used; while novelty doesn't rely on any common accuracy algorithm. These algorithms can be improved by pre-filtering, modeling and post-filtering.

Iaquinta et al. proposed introducing serendipity in a content-based recommender system creating consequently a hybrid recommender system that joins both, the content-based algorithm and the serendipitous heuristics (Iaquinta et al., 2008). According to them, the strategies to induce serendipity are as follows: implement it via "blind luck", i.e. randomly, or via user profile in what is called the Pasteur Principle, or via poor similarity measures, or even, via reasoning by analogy without any particular implementation. Therefore, some content-based recommender systems, like Dailylearner for instance, filter out the items that are too different, and also, too similar to the user's previously rated items.

The Pasteur Principle previously mentioned, as Pasteur himself states "chance favors only the prepared mind", was used by Gemmis et al. in their approach. The ability of the algorithm to produce serendipity can be improved by the knowledge infusion process (de Gemmis et al., 2015). Their study showed a better balance between relevance and unexpectedness, and that turned out to be better than other collaborative and content-based algorithms for recommendation. An interesting characteristic of their study was the measure of surprise done actively through the analysis of the users' face expressions. This analysis is performed using Noldus FaceReaderTM. That way, implicit feedback about the users' reactions will be gathered towards the recommendations that they are given. (de Gemmis et al., 2015)

In his model for news recommendations (Jenders et al., 2015), Jenders suggests many ranking algorithms and models and compares them. The serendipitous ranking uses a boosting algorithm to re-rank articles. Those articles are previously ranked according to an unexpectedness model and another model based on the cosine similarity between the items and a source article. This ranking system gained the highest mean surprise ratings per participant.

Reviglio in his study, states that serendipity cannot be created on demand (Reviglio, n.d.). Instead, it should be cultivated by creating opportunities for it. These opportunities would be present in a learning environment that can be physical or digital. He elaborates his concept through social media. He affirms that by pushing the user to burst from the bubble, we give the people the power to discover and by doing this, we create balance by giving freedom and mystery. As a continuation for what was previously said, Son et al. through their observation noted that microblogging communities provide a suitable context to observe the presence and effect of serendipity (Sun et al., n.d.). In fact, their experiment revealed a high ratio of serendipity due to retweeting. They remarked that this serendipitous diffusion of information affects the user's activity and engagement positively.

Some practitioners are trying to create systems where the design enhances serendipity. Two examples can be Google's theoretical serendipity engine and EBay's test in serendipitous shopping (Sun et al., n.d.). Another recommender framework that tries to introduce serendipity is Auralist (Zhang et al., 2012). This system attempts not only to balance between accuracy, diversity, novelty and serendipity in the recommendation of music, but also to improve them simultaneously. Observation of the systems reflects how users are ready willingly sacrifice some accuracy willingly to improve all the rest.

In order to better expect the unexpected, Adamopoulos et al. proposed a method to generate unexpected recommendations while maintaining accuracy (Adamopoulos and Tuzhilin, n.d.). We used their algorithm in our study, and therefore, we will be explaining it later.

3 IMPLEMENTATION ENVIRONMENT

In this section, we present the algorithm used followed by the dataset.

3.1 Strategies

In order to test the optimal number of serendipitous recommendations in the accurate list of recommendations, we started by choosing an algorithm for both our base strategy and the serendipity strategy. For the base strategy, we picked a non-personalized single-heuristic strategy. Our base study, which is supposed to generate accurate recommendations, is based on the popularity. In this strategy, the selection of the items is done in a descending order of popularity (i.e. number of ratings).

As for the serendipity strategy, which is personalized, it takes into consideration three factors in order to select the item and add it to the recommendation list: the quality, the unexpectedness and the utility. Certain restrictions and boundaries are placed in order to test if the item's quality is above a certain lower limit, and if it is farther enough from the expectations of the user (not too much, not too little).

Six cases were subject to our testing. In each case, we varied the number of recommendations generated by each of the two strategies previously mentioned. Starting from case one where all the items are generated by the base strategy, till the last case where all items are serendipitous, we changed the number of items as follows:

- Case 1: Strategy_10B_0S:

10 recommendations from the base strategy
No recommendation from the serendipity strategy

- Case 2: Strategy_8B_2S

8 recommendations from the base strategy
2 recommendations from the serendipity strategy

- Case 3: Strategy_6B_4S

6 recommendations from the base strategy
4 recommendations from the serendipity strategy

- Case 4: Strategy_4B_6S

4 recommendations from the base strategy
6 recommendations from the serendipity strategy

- Case 5: Strategy_2B_8S

2 recommendations from the base strategy
8 recommendations from the serendipity strategy

- Case 6: Strategy_0B_10S

No recommendation from the base strategy
10 recommendations from the serendipity strategy

3.1.1 Serendipity Algorithm

As we have previously mentioned, we used the algorithm implemented by Adamopoulos et al. (Adamopoulos and Tuzhilin, n.d.). Three main steps are used.

Step 1: Quality Calculations:
First, we fix a lower limit on the quality of the recommended items. The first test is a comparison between the item's quality and the lower limit. If its quality is higher, it continues to the next step.

Step 2: Unexpectedness Calculation:
The second step is to compute the set of expected recommendations Eu. Then, a lower limit on the distance of recommended items from expectations, and an upper limit are set. This is the range of unexpectedness. Once we compute the unexpectedness of a certain item, we check if it belongs to the range. Otherwise, the item is dropped from the recommendation list.

Step 3: Utility Calculation:
When the item passes the quality and unexpectedness tests, we need to estimate its utility for the user. The items with the highest utilities will be the ones recommended for the user in the end.
Considering that the study is done offline, the ratings of the users are used as a proxy for the utility of the recommendations.

3.1.2 Accuracy Algorithm

We used the algorithm implemented by Chaaya et al. (Chaaya et al., 2017), that was originally suggested by Elahi et al (Elahi et al., 2011).

R is our dataset. It is a matrix containing the items, the users and their ratings for some of the items. The user rating is presented by rui where i is the rated item by user u.

Four main steps are used in order to implement the accuracy algorithm.

Step 1: Dataset Partitioning
Divide R into three datasets in a random way:

- Dataset S (System): it contains the ratings known to the system that the user provided.
- Dataset Q (Queries): it contains the ratings for items unknown by the system, but that will be simulated from the user.
- Dataset E (Evaluation): as its name indicates, the purpose of this dataset is evaluation through the calculation of the accuracy.

A certain rating in the database will be present in one and only one of these three datasets (if the rating was not zero). In other words, there are no duplications. The not null ratings in R were divided randomly in the following percentages: around 0.5% in S, 69.5% in Q, and 30% in E. At the beginning, S contains very few ratings, reflecting what would happen in a real-life recommender system: the system possesses little information. This is the cold start problem that is faced by the recommender systems (Kunaver and Požrl, 2017).

Step 2: Rating Elicitation
We have the set Su that stands for System unknown. All the items that are not rated in S, for every user, are considered unknown information for the system. They will be placed inside Su. Through active learning, a certain number L among those items will be given to the user in order for him/her to rate the item in question. The ratings will be retrieved from the dataset Q. Afterwards, they will be transferred to S. Since there is no duplication, once those items are moved to S they will be removed from Q. No item will be rated twice by the user: all L items are removed from Su (System unknown). In the used algorithm, L is set to 10.

Step 3: Training Prediction Model
For every user in S the prediction model is trained. The objective of training the prediction model is to predict the ratings of the unrated items. Chaaya et al. used a neighborhood-based technique in order to predict the ratings (Chaaya et al., 2017).

First of all, the similarity between each two users is computed using Pearson correlation and summing over Iuv, the set of items rated by both users, u and v:

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r_u})(r_{vi} - \bar{r_v})}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r_u})^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r_v})^2}} \quad (1)$$

This value is then used in order to predict the ratings of the unrated items for user u, supposing that two similar users will rate the same item similarly. The predicted ratings rui are calculated using the following formula:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N_i(u)} sim(u,v)(r_{vi} - \bar{r}_v)}{\sum_{v \in N_{i(u)}} |sim(u,v)|} \quad (2)$$

(u) is the set of users similar to u and who rated the item i.

Step 4: Metrics Calculation
Many metrics exist in order to measure the success of the recommender system. Serendipity is deeply related to the user's satisfaction which is hard to measure or even define. Our experiment is done offline and is non-personalized. In other words, it does not include users. We will evaluate our technique using existing metrics. This is a common practice used when trying to evaluate the results, where the generated recommendations are compared with a baseline primitive recommendation system, and measurements are done through the use of saved ratings (Kaminskas and Bridge, 2016).

The evaluation was done using two predictive accuracy metrics: MAE and RSME. The Mean Absolute Error (MAE) computes the deviation between the actual and predicted ratings. Every prediction error is weighted in the same way.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i| \tag{3}$$

The Root Mean Square Error is similar to MAE; however, it places more emphasis on larger deviation.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2} \tag{4}$$

The MAE and RSME metrics are calculated on E. The algorithm then repeats the second, third and fourth step N times; N being the number of times every user logs in to the system. While repeating step three, the set Su is new and it should be considered.

3.2 Dataset

In this paper, we consider the 100K MovieLens dataset. It contains 100,000 ratings of 1682 movies. Those ratings were made by 943 users. A 5-point rating scale with the set {1, 2, 3, 4, 5} is considered. Every user has twenty ratings at least.

4 EXPERIMENTAL RESULTS

In this section we will compare the different strategies using the selected metrics. The graphs of Fig. 2 and 3 show the performance after every iteration from 1 to 10 for both, MAE and RSME. We limited our study for 10 iterations for many reasons. First, the size of the dataset is not very large, and the strategies tend to behave similarly after a certain period of time. Second, users tend to rate few items. Therefore, by limiting our iterations to 10 we are being more realistic.

The first observation is that the sixth case where all the items are recommended serendipitously performs the worst. This is expected and logical and was encountered by other researches (Chaaya et al., 2017). In fact, when all the items are serendipitous, the algorithm will behave identically to a random strategy, where accurate recommendations are not taken into consideration at all. Cases five and four have similarly bad results, since the number of serendipitous recommendation is still high. However, with case three we start seeing some better results. In the first three iterations, it still has a poor performance, but after that, it starts behaving almost the same as case one where all items are "supposedly" accurate. The first three cases are actually really close in performance. If we take a good look, strategy two has the best performance. A detailed table of the values resulting in each of the ten iterations for both metrics for every strategy is shown in Tables 1 and 2. Therefore, according to this study, and, in the given environment and conditions, eight accurate recommendations teamed with two serendipitous gave the best result.

The limitations on this study were many. Serendipity can be implemented using many algorithms and in different ways. Serendipity strongly affects the user's satisfaction which is already hard to understand or measure. An online study may be more relevant to how serendipity actually affects the recommendations. An implicit feedback is required for a better assessment, like in the work of Gemmis et al. where the facial expression was considered the key to measure surprise (de Gemmis et al., 2015). Moreover, the recommendation list size was fixed to ten which is not always the case. This goes without mentioning all the limitations that always occur in the recommender systems studies where many factors cannot be generalized and the results are restricted by the experiment itself.

5 CONCLUSION

Serendipity is an important factor in the recommender system that is still under construction. A clear definition is yet to be unified but what we can say for sure is that it is a happy surprise. The system is asked to predict the unpredictable, to expect the relevant unexpected. Many studies are

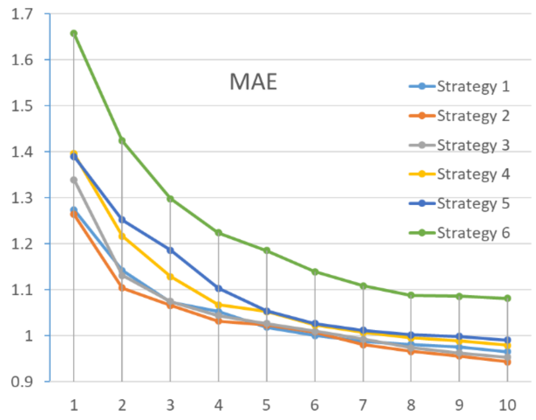

Figure 2: Evaluation of the strategies with MAE.

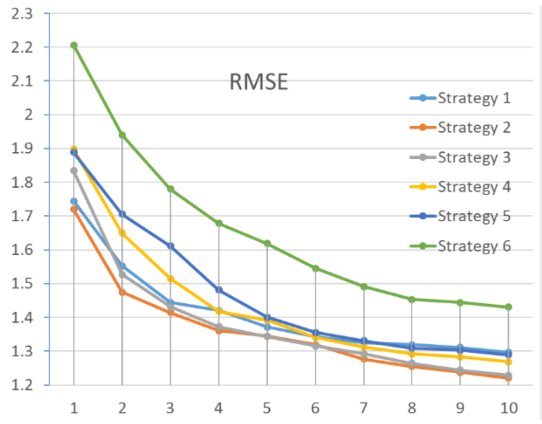

Figure 3: Evaluation of the strategies with RMSE.

Table 1: Detailed values of the evaluation of the strategies using MAE.

j	Strategy 1	Strategy 2	Strategy 3	Strategy 4	Strategy 5	Strategy 6
0	1.273995	1.264287	1.33942	1.396219	1.390011	1.656916
1	1.141812	1.104405	1.131474	1.21677	1.251889	1.42413
2	1.073603	1.066256	1.074856	1.129039	1.186332	1.298345
3	1.052768	1.032323	1.043951	1.067722	1.103232	1.223998
4	1.018811	1.023545	1.026679	1.052464	1.053776	1.185159
5	1.000731	1.007857	1.009808	1.023563	1.026923	1.139411
6	0.988289	0.980875	0.992785	1.007494	1.012334	1.108611
7	0.981468	0.966141	0.973857	0.995627	1.001796	1.088043
8	0.97515	0.956138	0.962973	0.988512	0.998751	1.086408
9	0.965468	0.943664	0.953171	0.97925	0.990942	1.081532

Table 2: Detailed values of the evaluation of the strategies using RMSE.

j	Strategy 1	Strategy 2	Strategy 3	Strategy 4	Strategy 5	Strategy 6
0	1.74410635	1.720034424	1.834427821	1.897848492	1.888841198	2.206197507
1	1.553062587	1.474470647	1.527057242	1.649357933	1.705949017	1.940098693
2	1.443750816	1.414175972	1.430911542	1.514621501	1.611318494	1.780124771
3	1.420222796	1.360419847	1.372571175	1.417455759	1.481355362	1.678748975
4	1.371658004	1.344541137	1.343920076	1.391354983	1.400491539	1.618303593
5	1.343840723	1.31899765	1.315324838	1.340393148	1.355270637	1.545510375
6	1.325739548	1.276450261	1.292058431	1.312050388	1.330443249	1.491448993
7	1.318660074	1.254746807	1.263709682	1.291586683	1.309292015	1.453229854
8	1.311383033	1.237695395	1.243987436	1.282841522	1.303148602	1.444063215
9	1.296779735	1.220012676	1.228174654	1.268312722	1.289391263	1.430291809

interested in finding a way to measure serendipity and, even more, to create it. In this paper, we proved that the presence of serendipity in the list of recommendations alongside some relevant recommendations will improve the user satisfaction. In the future, many improvements can be done to this study: new strategies can be proposed, different metrics can be used, and an online experiment can be conducted. Serendipity is a very vast world worthy of discovering and a face for recommender system that deserves to be invested in.

REFERENCES

Adamopoulos, P., Tuzhilin, A., n.d. On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected 50.

Chaaya, G., Metais, E., Abdo, J.B., Chiky, R., Demerjian, J., Barbar, K., 2017. Evaluating Non-personalized Single-Heuristic Active Learning Strategies for Collaborative Filtering Recommender Systems, in: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). Presented at the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, Cancun, Mexico, pp. 593–600. https://doi.org/10.1109/ICMLA.2017.00-96

de Gemmis, M., Lops, P., Semeraro, G., Musto, C., 2015. An investigation on the serendipity problem in recommender systems. Information Processing & Management 51, 695–717. https://doi.org/10.1016/j.ipm.2015.06.008

Elahi, M., Repsys, V., Ricci, F., 2011. Rating Elicitation Strategies for Collaborative Filtering, in: Huemer, C., Setzer, T. (Eds.), E-Commerce and Web Technologies. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 160–171. https://doi.org/10.1007/978-3-642-23014-1_14

Ge, M., Delgado-Battenfeld, C., Jannach, D., 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. ACM Press, p. 257. https://doi.org/10.1145/1864708.1864761

Iaquinta, L., Gemmis, M. de, Lops, P., Semeraro, G., Filannino, M., Molino, P., 2008. Introducing Serendipity in a Content-Based Recommender System. IEEE, pp. 168–173. https://doi.org/10.1109/HIS.2008.25

Jenders, M., Lindhauer, T., Kasneci, G., Krestel, R., Naumann, F., 2015. A Serendipity Model for News Recommendation, in: Hölldobler, S., Peñaloza, R., Rudolph, S. (Eds.), KI 2015: Advances in Artificial Intelligence. Springer International Publishing, Cham, pp. 111–123. https://doi.org/10.1007/978-3-319-24489-1_9

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 7, 1–42. https://doi.org/10.1145/2926720

Kotkov, D., Wang, S., Veijalainen, J., 2016. A survey of serendipity in recommender systems. Knowledge-Based Systems 111, 180–192. https://doi.org/10.1016/j.knosys.2016.08.014

Kunaver, M., Požrl, T., 2017. Diversity in recommender systems – A survey. Knowledge-Based Systems 123, 154–162. https://doi.org/10.1016/j.knosys.2017.02.009

Reviglio, U., n.d. Serendipity as a Design Principle for Social Media 5.

Sun, T., Zhang, M., Mei, Q., n.d. Unexpected Relevance: An Empirical Study of Serendipity in Retweets 10.

Zhang, Y.C., Séaghdha, D.Ó., Quercia, D., Jambor, T., 2012. Auralist: introducing serendipity into music recommendation. ACM Press, p. 13. https://doi.org/10.1145/2124295.2124300