Exploring coclustering for serendipity improvement in content-based recommendation

Andrei Martins Silva, Fernando Henrique da Silva Costa, Alexandra Katiuska Ramos Diaz and Sarajane Marques Peres

November 10, 2018

Exploring coclustering for serendipity improvement in content-based recommendation

Andrei Martins Silva[1], Fernando Henrique da Silva Costa[1], Alexandra Katiuska Ramos Diaz[1], and Sarajane Marques Peres[1][0000−0003−3551−6480]

School of Arts, Sciences and Humanities, University of São Paulo, Brazil
{andreimartins,fhscosta0993,katy.rd,sarajane}@usp.br

Abstract. Content-based recommender systems are widely used in different domains. However, they are usually inefficient to produce serendipitous recommendations. A recommendation is serendipitous if it is both relevant and unexpected. The literature indicates that one possibility of achieving serendipity in recommendations is to design them using partial similarities between items. From such intuition, coclustering can be explored to offer serendipitous recommendations to users. In this paper, we propose a coclustering-based approach to implement content-based recommendations. Experiments carried out on the MovieLens 2K dataset show that our approach is competitive in terms of serendipity.

Keywords: Content-based Recommender Systems · Serendipity · Coclustering · Nonnegative Matrix Factorization · Jaccard Similarity

1 Introduction

A recommender system (RS) helps users find items that are useful to them. These systems work by predicting relevant items to a user, given his previous interactions with the system, e.g: Amazon [13] recommends interesting products to its customers based on similar products they have bought, seen or liked before; Netflix [11] suggests movies based on its customers' watching history. Content-based recommendation is a successful approach for recommending items. The idea is to build a user profile from features which represents the items the user expressed interest before, compare that profile to unseen items and recommend the most similar ones [14]. Although widely adopted, this approach tends to present the problem of lack of serendipity, in which only very similar items are recommended to the user [1]. Serendipity is desirable because it allows the users to receive relevant and unexpected recommendations, which they would not be able to find autonomously [6]. Usually, content-based recommender systems (CRS) are based on global similarity - an item is similar to another regarding all their attributes. Coclustering allows finding partial similarities between items - a pair of items may be deemed similar even though the similarity occurs only over a subset of their attributes. Therefore, we propose the use of coclustering to find serendipitous recommendations. The underlying assumption is that by finding partial similarities, the adequate balance between relevance and unexpectedness can be achieved and, thus, serendipitous recommendations, provided.

This paper is organized as follows: section 2 provides some background on serendipity in the RS literature. Section 3 briefly describe the coclustering task and a special class of algorithms designed to solve this task. In section 4, our contribution, a coclustering-based approach to provide recommendations, is presented along with a second recommendation approach that is used for comparison purposes. Experimental results are presented in section 5. Sections 6 and 7 review some related works and conclude the paper.

2 Serendipity in recommender systems

Efforts to build good RSs were initially focused on reaching recommendation lists with high levels of accuracy [15]. Systems were designed to recommend items similar to those that users have liked in the past. However, recommendation strategies built on these ideas are limited because they suffer from a problem defined as over-specialization and are no longer sufficient to meet users' preferences. In fact, a RS must have mechanisms capable of recommending new items, different from those already known by the users and that meet their interests [7]. To satisfy these new needs, researchers in the RS field have worked with serendipity as an aspect to be sought when composing lists of recommendations.

In the context of RSs, serendipity is related to the quality of recommendations. In general terms, the recommendation is serendipitous if it brings relevant and unexpected items, i.e.: the items serve user's needs and are items that the users did not expect to receive; therefore, they would not have found them if they have solved their request on their own [6]. In the literature, the definitions for serendipity employ subjective terms and abstract notions that make serendipity a complex concept to understand and measure [7]. Some of these definitions are: *the experience of a user who has received an unexpected and fortuitous recommendation; how good an RS is at suggesting serendipitous items that are relevant, novel and unexpected for a particular user* [4]; *serendipitous items are, by definition, unpopular and significantly different from the user profile* [12].

The measurement of serendipity in RSs has received more attention from the scientific community in recent years [12]. Some efforts delegate the measurement of this aspect to procedures that directly involve the perception of the user [10], others propose quantitative measures based on the distance between the results produced by the method to be evaluated and those produced by a primitive prediction method [15, 4] and, finally, some authors propose measurements based on observations regarding the history of ratings and items popularity [5]. This latter strategy is adopted herein. Serendipity can be measured combining relevance and unexpectedness measures. To establish metrics for these concepts, the authors in [5] consider: a recommender system S with users u and items i; a recommendation list L with N items; τ_{ui} as the rating given to the item i by the user u; μ_u as the mean rating given by u to a subset of the items in S; an item i as relevant to a user u if $\tau_{ui} > \mu_u$; $\#\tau_i$ as the number of ratings given to the item i;

ν_S as the average number of ratings given to items in S; an item i as popular (or expected) in S if $\#\tau_i > \nu_S$[1], otherwise it is an unexpected item. Thus, relevance of L is defined as $\frac{\sum_{i \in L} R(i)}{N}$, where $R(i) = 1$ if $\tau_{ui} > \mu_u$ and $R(i) = 0$ otherwise. Unexpectedness of L is defined as $\frac{\sum_{i \in L} U(i)}{N}$, where $U(i) = 1$ if $\#\tau_i \leq \nu_S$ and $U(i) = 0$ otherwise. A recommendation is serendipitous if it is both relevant and unexpected. Regarding to L, serendipity is defined as $\frac{\sum_{i \in L} S(i)}{N}$, where $S(i) = 1$ if $(\tau_{ui} > \mu_i) \wedge (\#\tau_i \leq \nu_S)$ and $S(i) = 0$ otherwise.

3 Coclustering

Coclustering is a data mining task that allows the extraction of relevant and particular information from data. In coclustering, rows and columns of data matrices are simultaneously grouped, enabling the discovery of structures called coclusters. The coclustering task looks for coclusters that form a bi-partition (a partition of objects that is strongly related to a partition of attributes). Each cluster of objects is such that each object belonging to it is strongly and differently related to any other objects belonging to the same cluster with respect to all clusters of attributes and vice versa [16]. According to [9], coclustering methods have two main advantages: they are more effective in dealing with the curse of dimensionality problem and provide an insightful description of clusters of objects by associating clusters of attributes with clusters of objects.

Formally, coclustering is defined in [16] as follows: given a matrix $\mathcal{X} \in \mathbb{R}^{n \times m}$, in which n and m are respectively the number of lines and columns in the matrix, let x_{ij} be the element corresponding to line i and column j; $\overrightarrow{x_i}$ and $\overrightarrow{y_j}$ indicate the vectors associated respectively with line i and column j. A coclustering generates a bi-partition $\mathcal{C}_{k \times l}$ on $\mathcal{X}$ by producing a set of $k \times l$ coclusters, that is a partition $\mathcal{C}_r$ with k clusters of rows associated with a $\mathcal{C}_c$ partition with l clusters of columns. The bi-partition $\mathcal{C}_{k \times l}$ optimizes a given objective function.

Non-negative Matrix Factorization (NMF) is a class of algorithms designed to solve the coclustering task. It was studied as a method for data analysis able to extract knowledge about an object from the study of its parts. Later, researchers successfully applied NMF to extract useful information from textual data [18]. Orthogonal Non-negative Matrix Tri-Factorization [18], is a tri-factorization method that decomposes the original matrix $\mathcal{X} \in \mathbb{R}_{+}^{n \times m}$ into three new non-negative matrices, called "factors", $U \in \mathbb{R}_{+}^{n \times k}$, $S \in \mathbb{R}_{+}^{k \times l}$ and $V \in \mathbb{R}_{+}^{m \times l}$ (under certain orthogonality restrictions) by iteratively adjusting these factors according to updating rules towards minimizing objective function $J = \frac{1}{2} \parallel X - USV^T \parallel^2, \quad U^T U = I, \quad V^T V = I$, until it finds rows and columns partitions that best explain data. The adjustment rules are:

$$U = U \odot \frac{XVS^T}{USV^T X^T U}, \; V = V \odot \frac{X^T US}{VS^T U^T XV} \text{ and } S = S \odot \frac{U^T XV}{U^T USV^T V},$$

[1] Since $\#\tau_i$ is independent of the rating qualification, a "bad" item can still be popular.

where $\| \cdot \|^2$ is the *Frobenius* norm, U^T, S^T and V^T are the respective transposed matrices, $\odot$ is the Hadamard product and a sequence of matrices (e.g. USV) is the classic matrix product.

4 Content-based strategies for recommendation

We present two content-based recommender approaches: the first is purely based on the Jaccard similarity and follows a nearest neighbors fashion [2], which was chosen as the first baseline to be used for comparison purposes; the second, introduced herein, uses Jaccard similarity combined with information from co-clustering models obtained by applying ONMTF.

a) Jaccard similarity recommendation: The Jaccard index (J) is often used to determine similarity among sets. In the content-based recommendation scenario, an item i or a user profile UP_u is a set of n descriptive attributes, $I_i = \{att_1, att_2, \ldots, att_n\}$. If an object is represented as a set of attributes, say one as the set of attributes A and another one as the set of attributes B, one can easily use Jaccard index to calculate the similarity between such objects applying $J(A, B) = \frac{A \cap B}{A \cup B}$. This calculation represents the core of this recommendation approach. For an arbitrary user u and her respective set of items ratings, the strategy using Jaccard index is built as follows: (1) split the set of known ratings from u (Tr_u) into Tr_u^+ and Tr_u^- with $\tau_{ui} > \mu_u$ and $\tau_{ui} \leq \mu_u$, respectively; (2) build two sets of attributes, POS_u and NEG_u from items in Tr_u^+ and Tr_u^- to represent the attributes of items considered respectively positive and negative according to u; (3) build the set of attributes for the user profile: $UP_u = POS_u - NEG_u$; (4) build the set of attributes for each candidate item CI_u; (5) calculate $J(CI_u, UP_U)$ and (6) select top-N itens in CI_u with the highest scores for recommendation.

b) ONMTF-based recommendation: Matrix tri-factorization results in a model represented by the matrices U, S, V. When coclustering movies (cf. section 5), U and V provide information about clusters of movies and clusters of tags respectively, and S provide information on the relationship between clusters of movies and clusters of tags. We propose to incorporate all this information into a recommendation approach geared toward serendipitous recommendations. Figure 1 illustrates how coclustering supports the recommendation approach. Following Figure 1, to model user interests for an arbitrary user u, consider his set of movies (items) ratings, and: (1) determine the associations between movies and clusters of movies (M_{mc}) using information from matrix U; repeat the process with US to find associations between movies and clusters of tags (M_{tc}); (2) build two set of movies, the *liked movies* set - those for which $\tau_{ui} > \mu_u$ - and the *disliked movies* set - those for which $\tau_{ui} \leq \mu_u$ (cf. section 2); (3) establish a positive prototype of movie clusters associations (M_{mc}^+) and a positive prototype for tag clusters associations by averaging associations (M_{tc}^+) from *liked* movies; repeat the process to establish the negative counterparts (M_{mc}^- and M_{tc}^-) by averaging associations from *disliked* movies; (4) summarize the user profile (information about which cluster of movies a user likes the most and about the set of topics the user is most interested in by subtracting negative prototypes vectors from their corresponding positive prototype vectors; (5) associate such prototypes with movies and tags clusters models in order to establish the final user profile (UP_{mc} and UP_{tc}). After obtaining the user profile, the recommendation is a matter of finding movies which share content, in some level, with the user profile. This process is summarized as follows. For an arbitrary user u: (1) calculate $J(CM_{mc}, UP_{mc})$, where CM_{mc} is the association between candidate movies and clusters of movies; (2) calculate $J(CM_{tc}, UP_{tc})$ where CM_{tc} is the association between candidate movies and clusters of tags; (3) both similarities are averaged yielding scores for candidate movies; (4) truncate[2] the list of candidates in the score $J = 0.25$ and, (5) from the survivor candidates, select the top-N highest scored movies for recommendation.

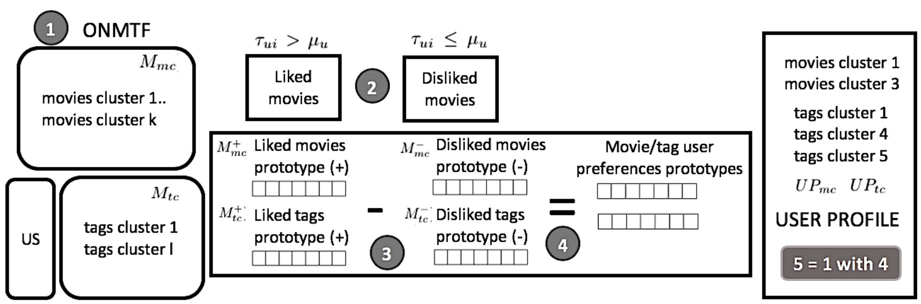

Fig. 1. Overview of coclustering for user interests modeling

5 Experiments, results and discussions

In this section, we describe the procedures for evaluating the recommendation approaches' performance in terms of three measures (cf. section 2), and present an example of recommendation list obtained from ONMTF-based approach.

5.1 Dataset and vector representation

The MovieLens 2K dataset[3] [3], used in the experiments, is a dataset with metadata of movies and anonymous user ratings for movies. Movies are items that must be recommended by the system S in the experiments. Tags are assigned to

[2] Since similarities among movies and tags have already been treated in the coclustering process, the maximization of J would insert an overspecialization in the process. The threshold score (0.25) was obtained empirically from extensive tests.

[3] The original Movielens dataset is provided by GroupLens research group (http://www.grouplens.org). In this study, the following files were used: `movies.dat`, `movie_tags.dat`, `tags.dat` and `user_ratedmovies-timestamps.dat`.

movies by users so that each tag may have been assigned zero or multiple times. Users express their judgment by assigning ratings to movies in a ten-point scale ranging from 0.5 to 5 with 0.5 steps. Some statistics about MovieLens 2K dataset are: 2,113 users, 20,197 movies, 20 genres, 13,222 tags, 855,598 ratings, 22,696 tags per user (avg), 8,117 tags per movie (avg), 2,040 genres per movie (avg), 404,921 ratings per user (avg) and 84,637 ratings per movie (avg).

Preprocessing procedures were performed on the dataset. Such procedures involved the choice of a subset of movies, a subset of tags used as movies descriptors and a subset of users. Only tags associated with movies by more than one user were retained, since we considered that tags assigned only once to a movie carries no relevant meaning. There are movies with no meaningful tags assigned to it, and they were discarded from the original dataset. The final subset was composed by 2,004 movies and 1,101 tags. Only users who evaluated more than 25 movies (1,967 users) were maintained so that a cross-validation procedure could be performed in the experiment. A vector representation for movies was built following a vector space model in which movies are vectors and tags are features. Thus, the dataset was transformed in a movies/tags matrix $\mathcal{X}^{2004\times1101}$, with cells filled in according to presence or absence of a tag assigned to a movie, i.e., $\overrightarrow{x_i} = \{w_{i,1}, \ldots, w_{i,j}, \ldots w_{i,M}\}$ where $\overrightarrow{x_i}$ is a movie, N is the number of movies, $i = \{1 \ldots N\}$, M is the number of tags, $j = 1 \ldots M$ and $w_{i,j} \in \{0, 1\}$, depending on whether the tag j is assigned to the movie i.

5.2 Procedures and results

To analyze the recommendation approaches, recommendation lists were generated and evaluated through a procedure inspired in those carried out in [5] and composed by six steps: (1) the subset of movies rated by u, $\mathcal{X}_u$, is split into five folds of movies; (2) the user profile is built from four folds (training set); (3) the lists L_a and L_b, with five items, are built by applying the recommendation approaches to select the most appropriate candidate movies from the remaining fold (test set); (4) L_a and L_b are evaluated in terms of relevance, unexpectedness and serendipity (cf. section 2); (5) the steps 2, 3 and 4 are repeated for each of the five folds. Results are averaged for each metric, and (6) common measures of position are extracted from 1,967 repetitions of the whole aforementioned procedure (one execution for each user). Wilcoxon test is run over final results to verify whether or not the means are significantly different.

Relevance, unexpectedness and serendipity were calculated for both recommendation approaches. Table 1 shows the distribution of all quality scores. According to relevance measures, the recommendation strategy based on Jaccard index yields better scores, with $p < 0.001$ in Wilcoxon test for *mean*. This result implies that, generally, Jaccard-based approach recommends movies that meet the interests of users more accurately, i.e., such recommendations suggest items whose rating would be higher than the average rating for that user. Even though ONMTF achieved lower relevance scores, it also provides fairly accurate recommendations most of the time, i.e., at least three relevant items out of five items in the recommendation list, in 50% of the time.

Table 1. Quality of recommendations

	Relevance		Unexpectedness		Serendipity	
	Jaccard	ONMTF	Jaccard	ONMTF	Jaccard	ONMTF
Min.	0.1600	0.1600	0.0000	0.0000	0.0000	0.0000
1st Qu.	0.6400	0.5200	0.0400	0.1200	0.0000	0.0400
Median	0.7200	0.6400	0.0800	0.2000	0.0400	0.0851
Mean	**0.7194**	0.6211	0.1122	**0.2105**	0.1235	**0.1673**
3rd Qu.	0.8400	0.7200	0.1600	0.3167	0.1167	0.2000
Max.	1.0000	0.9600	0.7200	0.7600	1.0000	1.0000

Regarding the evaluation of unexpectedness, the results in Table 1 shows that the ONMTF-based approach overcomes that based on Jaccard, with $p < 0.001$ in Wilcoxon test. In 50% of the recommendation lists, ONMTF-based approach provides at least one unexpected item while Jaccard recommends one unexpected item in approximately 12% of the time. When it comes to serendipity, ONMTF-based approach also surpass Jaccard-based approach, with $p < 0.001$ in Wilcoxon test. Most of the time, both approaches struggle to offer serendipitous recommendations. However, ONMTF-based approach shows at least one serendipitous recommendation 25% of the time, while Jaccard-based approach shows at least one serendipitous item in 17% of the time.

5.3 Analysis of a recommendation list

According to the results presented in section 5.2, the ONMTF-based approach is capable of contributing to the generation of serendipitous recommendations at least once every five recommendations. Since there is clearly room for improvement, it is desirable to better understand how a serendipitous recommendation can be achieved. For this purpose, consider the example related to a specific user who has requested a recommendation from the system S.

Movies are split and combined into prototype vectors according to positive and negative reviews (cf. section 4). These vectors reveal user interests in terms of genres and topics. User 11114 topic preferences (extracted from the factor US) and genre interests (extracted from factor U) are shown in the figures 2 and 3, respectively. In the figure 2, the size of words corresponds to the relevance of topics to that user (the higher the value in US is, the more appropriate to represent a subset of movies a tag cluster is, and the more frequent in these movies a tag from such tag cluster is, the bigger the tag is in the cloud). In figure 3, each cluster is represented by its six more representative movies (the higher the value in U is, the more representative for the cluster the movie is). The colored dots indicate the genre associate to each cluster. Clusters were labeled based on genre most often associated with their six more representative movies.

The recommendation approach takes the captured interests as a basis and search for movies similar to the user profile. As described before, in order to balance relevance and unexpectedness, the strategy looks for moderate resemblance (between 0% and 25%) instead of maximize resemblance between candi-

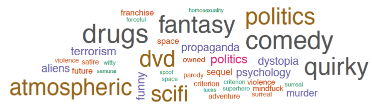

Fig. 2. User 11114: Topic interests

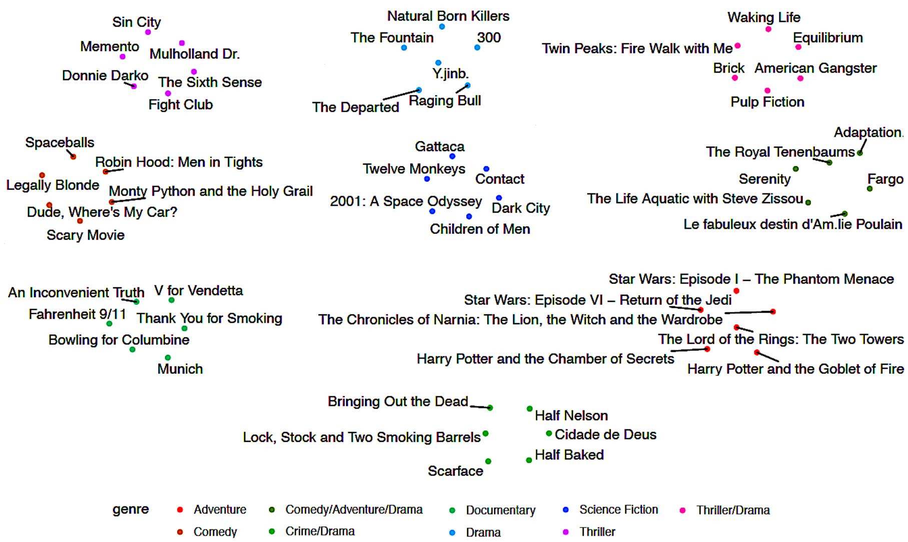

Fig. 3. User 11114: Genre interests

date movies and user profile. Finally, the recommendation list is composed by the top-N most adequate items according to such resemblance. The recommendation list produced for the user 11114, the rating assigned to each item by the user and the quality reached by each recommended item are shown in table 2.

Table 2. Recommendation list for user 11114

Movie title	Rating	Genre	Relevance	Unexpectedness	Serendipity
Waiting...	4.0	Comedy	x	x	x
Juno	4.5	Comedy	x		
Spy Kids 2	1.0	Adventure		x	
The Matrix Revolutions	2.5	Adventure			

The first recommendation - Waiting... - is a serendipitous recommendation. It adheres to the user profile as it features elements of comedy and it is also an unexpected recommendation (this is not a popular item in the system S). Juno is also aligned to the user profile, but is rather popular and thus, not an unexpected recommendation. Spy Kids 2 has elements of the user profile such as adventure and comedy veins and it is also unexpected, however it fails to meet the relevance criteria. Even though Matrix Revolutions shows some resemblance with the user profile (adventure, action, thriller, science fiction, dystopia, future, psychology), however it fails to meet both relevance and unexpectedness criteria.

6 Related works

Much attention has been recently brought to the serendipity on RS. In [12], the authors outlined the state of the art in serendipity for RS. They suggested that serendipity-oriented algorithms and evaluation metrics should take into account both item popularity and similarity to a user profile. They highlighted the potential of context-aware and cross-domain RS for serendipitous recommendations, since such approaches can make use of additional information rather than just the user preferences. In [17], the authors defended the extraction of interests from user activity on Twitter to suggest serendipitous connections. Their algorithm extracts about 11% of serendipitous terms from user activity. They concluded that extractions from user's tweets are more likely to extract relevant terms, and the enrichment from web pages can bring the unexpectedness component for serendipity connections. In [10], authors presented a model that combines the cosine similarity and an unexpectedness model that recommends serendipitous news articles. A museum tour recommender is presented in [8], where the authors proposed a hybrid RS that combines a content-based approach and serendipity heuristics to provide serendipitous artwork recommendations.

In the movie recommendations context, a framework was developed by [19] which balance degrees of relevance and surprise in recommendations. Knowledge infusion process into a random walk algorithm was proposed in [5] in order to produce serendipitous recommendations. They conducted an *in vitro* experiment similar to the experiment described in this paper and achieved on average, 15% of serendipitous items in recommendation lists.

7 Conclusion

In this paper, we introduced the use of coclustering for producing serendipitous recommendations in CRSs. Experimental results showed that our approach can produce serendipitous recommendations, achieving one serendipitous recommendation 25% of the time. However, the lack of serendipity in CRSs is still an open research question, mainly because it is difficult to generate recommendations out of the obvious path, and that still remain relevant to the target user. Related works that have achieved advances in serendipitous recommendations reports success rates about 10% to 15%. At present, direct comparisons between our approach and those presented in related works are not feasible, since the conditions of experimentation are not fully compatible. In the future, we intend to explore the tri-factorization model in other ways to improve recommendations and allow direct comparisons. Besides, evaluation with user studies are planned.

References

1. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Communications of the ACM **40**(3), 66–72 (1997)
2. Billsus, D., Pazzani, M.J.: User modeling for adaptive news access. User Modeling and User-Adapted Interaction **10**(2-3), 147–180 (2000)
3. Cantador, I., Brusilovsky, P.L., Kuflik, T.: 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec2011). ACM (2011)
4. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: Proc. of the 4th ACM Conf. on Recommender Systems. pp. 257–260. ACM (2010)
5. de Gemmis, M., Lops, P., Semeraro, G., Musto, C.: An investigation on the serendipity problem in recommender systems. Information Processing & Management **51**(5), 695–717 (2015)
6. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. on Inf. Syst. **22**(1), 5–53 (2004)
7. Iaquinta, L., De Gemmis, M., Lops, P., Semeraro, G., Filannino, M., Molino, P.: Introducing serendipity in a content-based recommender system. In: 8th Int. Conf. on Hybrid Intelligent Systems (HIS'08). pp. 168–173. IEEE (2008)
8. Iaquinta, L., de Gemmis, M., Lops, P., Semeraro, G., Molino, P.: Serendipitous encounters along dynamically personalized museum tours. In: Proc. of the 1st Italian Information Retrieval Workshop (IIR'10). pp. 101–102 (2010)
9. Ienco, D., Robardet, C., Pensa, R.G., Meo, R.: Parameter-less co-clustering for star-structured heterogeneous data. Data Mining and Knowledge Discovery **26**(2), 217–254 (2013)
10. Jenders, M., Lindhauer, T., Kasneci, G., Krestel, R., Naumann, F.: A serendipity model for news recommendation. In: Joint German/Austrian Conf. on Artificial Intelligence. pp. 111–123. Springer (2015)
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8) (2009)
12. Kotkov, D., Wang, S., Veijalainen, J.: A survey of serendipity in recommender systems. Knowledge-Based Systems **111**, 180–192 (2016)
13. Linden, G., Smith, B., York, J.: Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet computing **7**(1), 76–80 (2003)
14. Lops, P., De Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Recommender Syst. Handbook, pp. 73–105 (2011)
15. Murakami, T., Mori, K., Orihara, R.: Metrics for evaluating the serendipity of recommendation lists. In: Annual Conf. of the Japanese Society for Artificial Intelligence. pp. 40–46. Springer (2007)
16. Pensa, R.G., Boulicaut, J.F., Cordero, F., Atzori, M.: Co-clustering numerical data under user-defined constraints. Stat. Analysis and Data Mining **3**(1), 38–55 (2010)
17. Piao, S., Whittle, J.: A feasibility study on extracting twitter users' interests using NLP tools for serendipitous connections. In: 3rd Int. Conf. on Privacy, Security, Risk and Trust and 3rd Int. Conf. on Social Computing. pp. 910–915. IEEE (2011)
18. Yoo, J., Choi, S.: Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds. Information Processing & Management **46**(5), 559–570 (2010)
19. Zheng, Q., Ip, H.H.: Customizable surprising recommendation based on the tradeoff between genre difference and genre similarity. In: Int. Conf. on WEB Intelligence and Intelligent Agent Technology. vol. 1, pp. 702–709. IEEE (2012)