IERG4999-VJ01 Graduation Thesis II

Introducing Serendipity in Recommender Systems

By:
WANG, Tianming 1155029084

A FINAL YEAR PROJECT REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF BACHELOR OF INFORMATION ENGINEERING
DEPARTMENT OF INFORMATION ENGINEERING
THE CHINESE UNIVERSITY OF HONG KONG

May, 2017

Introducing Serendipity in Recommender Systems

Tianming Wang

Abstract

Recommender systems are filters that suggest possibly interesting items to users. The systems discover one's potential interests by analyzing his past behaviors. However, traditional recommendation algorithms mainly focus on recommending items similar to the ones liked by the user before, causing lack of novelty in the suggestions. In this project, we investigate this issue, known as overspecialization or serendipity problem, by proposing and implementing a content-based recommendation algorithm with back-ground knowledge that enables the system to deeply understand characteristics of items and user preferences. The algorithm is evaluated by two experiments: an offline experiment on HETREC2011-MOVIELENS-2K database and a preliminary user study performed using Amazon Mechanical Turk service. These experiments show that the proposed algorithm promotes non-obvious suggestions, while minimizing the accuracy loss.

Keywords: Recommender Systems, Serendipity problem

1. Introduction

Recommender systems, with its capability in suggesting users interesting objects from large number of possible options and hence increasing corporate profits, is an emerging research topic in recent years, currently, there are two major types of recommender systems.

User based systems, i.e. user-based collaborative filtering, which looks for users that share the same interests (similar rating results) with the target and use the ratings from these like-minded ones to predict and recommend. While ensuring rather high level of relevance, this approach suffers from scalability issue: as the number of customers and items increases, the computation time of algorithms grows exponentially (Yao, 2015).

Content based systems, i.e. item-based collaborative filtering, which utilizes a series of discrete characteristics of an item to recommend additional ones that are like those users has liked in the past. This algorithm was introduced to overcome the scalability issue of user based collaborative filtering as they compute item similarities in an offline manner, but this method has one significant limitation: the recommendations generated can be obvious to the user, as they are close to the items that the user has already known before, this issue is called over-specialization problem (Yao, 2015).

1.1 Definition of over-specialization

Over-specialization problem is one of the major open issues across all recommendation algorithms, especially content-based ones. Content-based filters tend to produce recommendations with a limited degree of novelty, because they try to find out the most relevant items based on user past behaviors. For example, if a user has liked action movies before, movies recommended to her based on content based filters will most likely be action ones, while the user may have other interests. To improve user satisfaction, recommender systems should be able to make serendipitous suggestions.

1.2 Concepts of serendipity

Serendipity is a difficult concept to study, as it includes an emotional dimension. One of the most widely-accepted definitions is proposed by Corneli et al., suggesting focus shift as a key condition for serendipity, which happens when something initially perceived irrelevant, neutral or even negative becomes relevant. Therefore, a serendipitous item must be unexpected and relevant to create a focus shift. Moreover, the user must not be familiar with the item (the item is novel), so she will pay attention to it. The relations among different items sets of familiar, novel, rated (movies that the user has given ratings for), relevant, unexpected and serendipitous from the perspective of the user is illustrated in the Figure 1 (Wang, 2016).

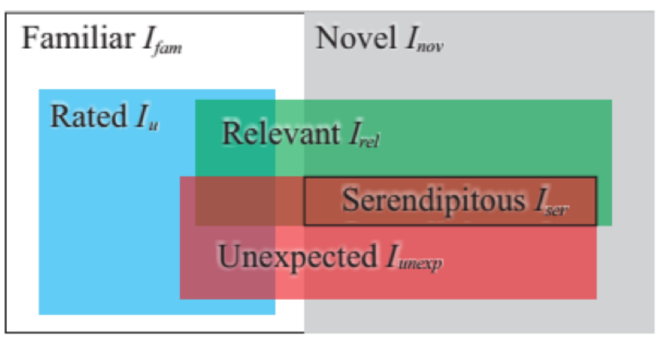


Figure 1: Serendipitous items must be novel, unexpected and relevant to the user at the same time. (Picture taken from Wang et al. 2016.)

1.3 Related work

In this project, we aim to propose a serendipitous recommendation algorithm and build an online movie recommender system base on that. Thus, we review related serendipitous recommendation algorithms and discuss the procedures that how they are related to our work.

Last semester, we studied and implemented a graph-based algorithm, named RWR-KI, proposed by Gemmis et al. in 2015. Gemmic's algorithm exploits a knowledge infusion process, which builds a computer-understandable knowledge repository fed by information from Wikipedia and WordNet. He also builds a Spreading Activation Network (SAN) from previous knowledge repositories. The recommendation step is based on hidden information retrieved from the Spreading Activation Network.Gemmic also conducted a user study to prove the accuracy of his algorithm.

This semester, we studied another serendipitous mobile app recommendation algorithm proposed by Bhandari, Sugiyama, Datta, & Jindal in 2013. They firstly build an app-app similarity graph where only similar apps are connected by an edge. Later, they use the list of apps in user's mobiles as starting and ending nodes, fetching shortest paths in the graph and recommend apps along the paths. We borrow Bhandari's idea in the second step of our algorithm.

2. Methodology

2.1 Inspiration and architecture

As described in section 1.2, in order for a recommendation to be serendipitous, it must be unexpected and relevant at the same time. In other words, we need to make recommendations satisfying user's potential interests. In the concept of user modeling, each user is associated with two types of interests: implicit and explicit. Explicit interests are directly collected from user past behaviors, e.g. purchase record, movie download history, and implicit ones are those users are not aware of. Traditional content-based recommender systems mainly focus on user's explicit interests, while serendipitous recommendations should base on those implicit ones.

Our proposed serendipitous recommendation algorithm consists of four major parts:

- M1: Similarity calculation
- M2: Feature network construction
- M3: User profile learning
- M4: Recommendation

In M1, we compute similarity value between each pair of movies based on their movie tags. In M2, we take a similar approach as the mobile app recommendation algorithm by Bhandari to build a similarity graph of movie items. In M3, we construct movie and user implicit profiles with the graph and make recommendations in M4. Details of each step are described in section 2.2.

2.2 Implementation details[1]

M1: Similarity calculation

In our movie database, each movie is associated with a list of movie tags (words and phrases) and we represent movies using these tags by Vector Space Model (VSM).

[1] this section is co-referenced with FYP partner Jinwen Hu

Word2Vec model, a computationally-efficient predictive model for Natural Language processing, is a common tool for language processing. The Word2Vec model takes in a large text corpus such as Wikipedia, produces a continuous vector space, where semantically similar words and phrases are mapped to nearby points. In particular, we adopt the pre-trained model published by Google on part of Google News dataset (about 100 billion words) for calculating similarity.

After representing movies by the Vector Space Model, similarity between movies is defined as the conventional consine similarity between their representative vectors.

M2: Feature Network Construction

This step takes in the similarity correlation matrix derived in $M1$ and build the feature network. Assume $G = (V, E)$ be an weighted and undirectional graph where V is the set of and E are a set of features and a set of edges, respectively. Our approach creates an edge between two features if the similarity between them is greater than a predefined threshold, weighted by the similarity score. The output is a feature-feature similarity graph that has vertices with the feature IDs, and edges with similarity scores between the vertices. Figure 2(a) shows the apps as vertices and edges (solid line) that connect the vertices marked with similarity between the apps (Bhandari, 2013).

M3: User profile learning

Movie tags and tag weights provided by IMDB database are adopted to produce feature vectors, and each movie is represented by a feature vector of length L:

$$f_i^m = (w_{i,1}^m, w_{i,2}^m, \cdots, w_{i,L}^m)$$

where L is the total number of movie tags in the database, $w_{i,j}^m$ is the tag weight of tag j with respect to movie item i.

Users are represented in a similar method, except that every user has 2 feature vectors, one is for explicit interests and the other is for implicit interests. Explicit feature vectors are averaged movie vectors weighted by their rating results. s user ratings and explicit and implicit profiles of movies, which are weighted averages of the respective profiles of the user rating matrix $R^{M \times N}$:

$$f_j^{u,ex} = \sum_{R_{j,i}>0} (R_{j,i} - 3)\, f_i^m$$

where $R_{j,i}$ is the rating result of user j on movie item i.

Feature network is applied in the process of generating implicit vectors. For each user, we treat his explicit features as starting and ending nodes in the feature network. After fetching the shortest paths between every pair of nodes, we use those entries along the path to construct user's implicit vector. The weight of feature k with respect to user j is computed as follows:

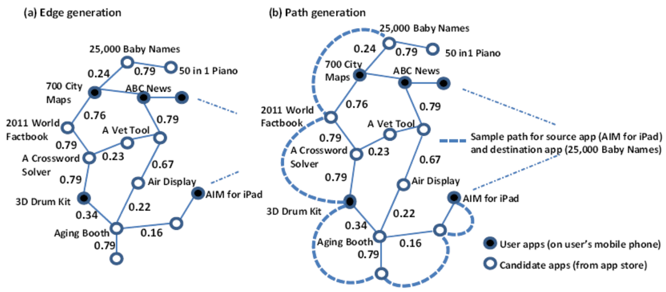


Figure 2: app network edge and path generation (Picture taken from Bhandari. 2013)

$$w_{j,k}^{u,im} \quad = \quad \sum_{(s,p)\in SP_j} (f_{j,s}^{u,ex} + f_{j,d}^{u,ex})/2$$

where SP_j is the set of possible short paths between, (s, p) is the starting and ending nodes of a path in SP_j. The aim of shortest-path is to reduce the overall cost to traverse from a given source node to its destination. Paths are constructed for each pair of features in user's explicit profile. Feature pairs with no path connecting them are eliminated due to lack of transitive relation between them. The path construction step represents the discovery of new interests and is useful for generating candidates for serendipitous recommendations. This procedure is shown in Figure 2(b)

M4: Recommendations

This step takes movie explicit profiles and user implicit profiles and aims to make recommendations base on that. We firstly construct a correlation matrix $C^{M\times N}$ of shape $M \times N$ that represents cosine similarity between movies' explicit profiles and users' implicit profiles. Then we construct a score matrix $S_{M\times N}$ based on C with entry $S_{j,i}$ scaled by the average rating on movie i :

$$S_{j,i} \quad = \quad C_{j,i} \times avgr(i)$$

Top-k movie in the score matrix is then selected for each user as the recommendation list.

3. Experiment

The main goal of the experiment is to validate the hypothesis that the top-k recommendation results produced by Content-based recommendation enhanced by feature network (CBR-FN) are serendipitous. Measuring serendipity levels is difficult as it requires objective measurement of emotional responses. To make our task more tractable, we consider serendipitous suggestions to be those that are relevant, i.e. close to user profiles, and unexpected at the same time.

While computing relevance is relatively easy, unexpectedness can be measured from different approaches. Therefore, we designed two experiments and compare the performance of our algorithm with some state-of-art recommendation algorithms.

3.1 Data pre-processing

The experiments are based on a subset of the HETREC2011-MOVIELENS-2K dataset available on Kaggle website, the original dataset contains 855,598 rating assignments on 10,197 movies given by 2113 users on a 10-point scale from 0.5 to 5. However, to facilitate the process of online experiment which needs at least the plot keywords and description for each movie, we had to discard the set of movies that failed to provide required information, resulting in a final list of 7688 movies distributed in 20 categories.

3.2 Baseline method

We compare the results of our strategy to the following algorithms:

Random: This baseline method randomly selects N movies from the database and the purpose of introducing this algorithm is to compare the performance of our algorithm to the blind luck strategy (Toms, 2000).

User to user collaborative filtering (U2U-CF): collaborative filtering is the most commonly used recommendation algorithms adopted by many big firms such as Amazon and Google. Based on a user-movie rating matrix, this method calculates similarities between users, makes predictions and recommends according to these calculated similarity values. Collaborative filtering is effective in generating accurate (relevant) recommendations.

3.3 Offline test

The main purpose of the offline experiment is to study the trade-off between relevance and unexpectedness of recommendation algorithms, we are eager to learn whether it is possible to introduce novelty to recommendation results and to ensure rather high level of accuracy at the same time. Results produced by the proposed CBR-FN are compared with those generated by baseline methods as described in section 3.2. Furthermore, we compute the relevance, unexpectedness and serendipitous levels by metrics described in section 3.3.1.

3.3.1 Metrics

Relevance can be defined in a binary manner: an item is either liked or disliked by the user. Therefore, we define an item i to be relevant to a user u if the rating result given by u on i is higher than or equal to the average rating given by the user. As result, the relevance level of a given recommendation list L of size N can be defined as the ratio between the size of the subset of relevant items to N (Gemmis et al. 2015):

$$\text{Relevance}(N) = \frac{\sum_{i \in L} R(i)}{N}$$

where:

$$R(i) = \begin{cases} 1 & \text{if } i \text{ is relevant;} \\ 0 & \text{otherwise.} \end{cases}$$

On the other hand, whether an item is unexpected or not is harder to define as it lies in the scope of emotional measurement. In our test, every user and movie is associated with a profile vector of equal length, thus the unexpected level of a given recommendation list L with respect to user u can be defined as the cosine distance $(1 - cosinesimilarity)$ between the profile vector of u and averaged profile vectors of L.

$$\text{Unexpectedness}(N) = 1 - cosinesimilarity\,(u, L)$$

Alternatively, unexpectedness can also be partially described by popularity score. An item i is defined to be popular if the number of users who have rated for i is higher than the average rating number across all items. As result, the popularity level of a given recommendation list L of size N can be defined as the ratio between the size of the subset of popular items to N (Gemmis et al. 2015):

$$\text{Popularity}(N) = \frac{\sum_{i \in L} P(i)}{N}$$

where:

$$P(i) = \begin{cases} 1 & \text{if } i \text{ is popular;} \\ 0 & \text{otherwise.} \end{cases}$$

With these two definitions, we define serendipity level of an given recommendation list L of size N to be the ratio between the size of the subset of serendipitous items (i.e. items that are both relevant and unexpected) to N (Gemmis et al. 2015):

$$\text{Serendipity}(N) = \frac{\sum_{i \in L} S(i)}{N}$$

where:

$$S(i) = \begin{cases} 1 & \text{if } i \text{ is serendipitous;} \\ 0 & \text{otherwise.} \end{cases}$$

3.3.2 Procedure

1. Build Word2Vec model with google pre-trained dataset "GoogleNews-vectors-negative300.bin" which contains more than 3 million words and phrases, and construct a similarity correlation matrix S for movie tags
2. Build feature network N based on similarity correlation matrix S
3. Construct explicit and implicit movie profiles
4. Construct explicit and implicit user profiles as weighted (rating) average of explicit and implicit movie profiles
5. Recommend 50 movies based on similarities between user implicit profiles and movie explicit profiles (ignore users that have rated less than 50 movies)
6. Performance measurements are conducted on recommendation lists for all users

3.3.3 Discussion of results

We adopt the metrics described in section 3.3.1 to compute the relevance, unexpectedness and popularity levels of the 50-movie recommendation list per user, and Table 1 presents the averaged results. The statistics presented in Table 1 shows that CBR-FN significantly outperforms U2U-CF and RANDOM algorithms in terms of unexpectedness, but its relevance level is slightly lower than U2U-CF. Therefore, CBR-FN is more balanced than U2U-CF and RANDOM algorithms and is the best algorithm among these three for recommender systems that focus equally on relevance and unexpectedness of recommendation results.

Table 1: General statistics on user feedback.

Metrics	CBR-FN	U2U-CF	RANDOM
Relevance	0.61	0.63	0.57
Unexpectedness	0.62	0.42	0.45
Popularity	0.59	0.59	0.59

3.4 Online test

Last semester, we built an online movie recommendation system and invited 40 Master (experienced and creditable workers) from Amazon Mechanical Turk platform to participate and give us feedback on the movies recommended by RWR-KI and RANDOM algorithms. This semester, we further improve the online system to include our proposed CBR-FN and U2U-CF algorithms and invite another 80 Master to do the online experiment. The main purpose is to study the acceptance of recommendation results produced by the CBR-FN algorithm in real world.

3.4.1 Procedure

In phase one, users are asked to select 3 to 5 favorite movie genres and are later shown 20 most popular movies from selected categories (movies with the highest number of viewers). For each movie, users are asked to rate based on a 5 point scale (1 is for most dislike and 5 is for most like) (Figure 3). The collected 20 rating results are used to construct user implicit profiles and 5 movies with similarities will be generated as recommendations. These 5 movies, together with another 5 movies generated by U2U-CF are given as the final recommendation list to the user.

In phase two, recommended items are shown one by one to the user (Figure 4). Meanwhile, two questions concerning unexpectedness and relevance levels will be asked: “Do you know this movie before?", and “How much do you like the movie?". Trailers and movie descriptions are presented in each page so that the user can be better informed to answer the questions. The answer to the first question was a measurement of unexpectedness level and the answer to the second question was interpreted as how relevant recommended results were.

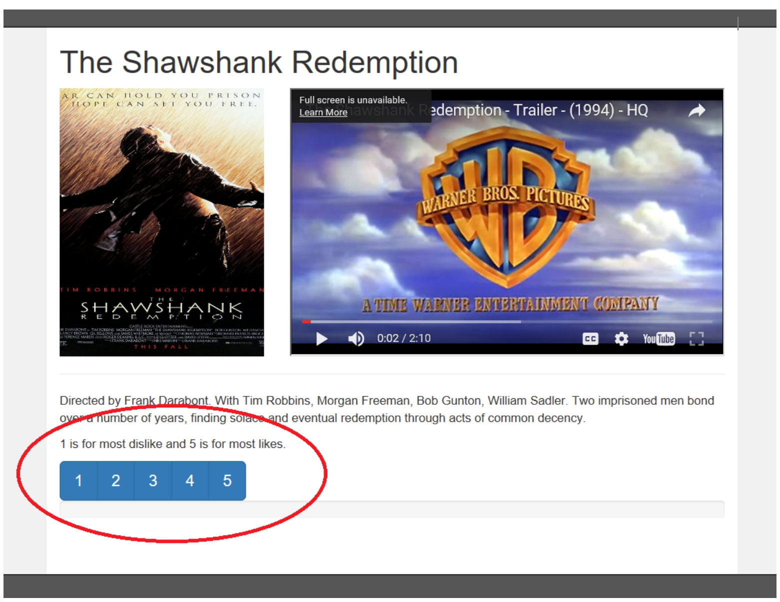


Figure 3: Users will give ratings to 20 movies in phase 1.

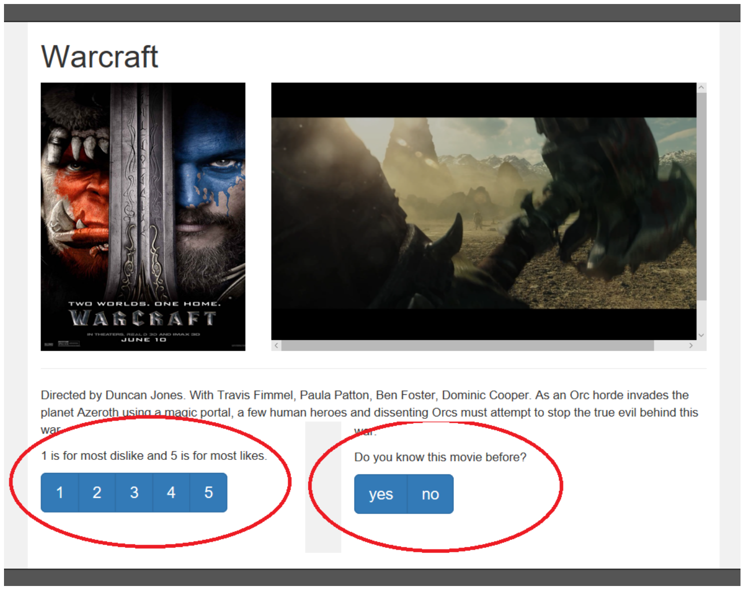


Figure 4: Users will answer two questions for each recommendation in phase 2.

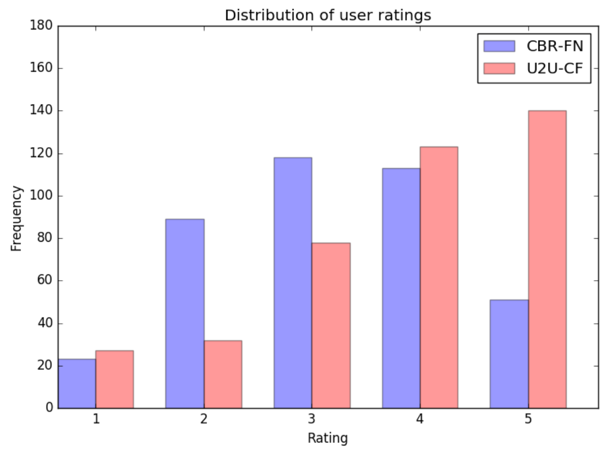
Table 2: General statistics on user feedback.

Metrics	CBR-FN	U2U-CF	RWR-KI	RANDOM
Relevance	0.77	0.85	0.63	0.59
Unexpectedness	0.80	0.07	0.60	0.54
Serendipity	0.59	0.04	0.35	0.27

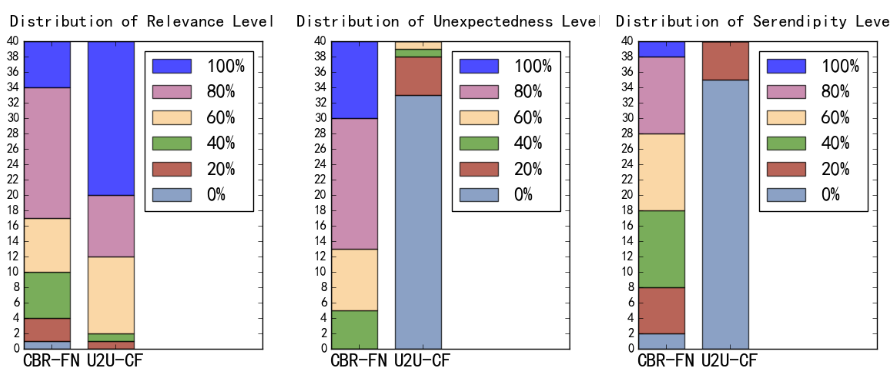
3.4.2 Analysis

We adopt a similar approach as in section 3.3.1 to compute the relevance, unexpectedness and serendipity levels of each 5-item recommendation list, and they serve as the performance measures of the recommendation algorithms. The statistics presented in Table 2 shows that CBR-FN significantly outperforms the other three baseline algorithms in terms of unexpectedness and serendipity, but its relevance level is slightly lower than U2U-CF in relevance, although the distribution of user ratings does not distinguish the two algorithms substantially (Figure 5(a)). RWR-KI slightly outperforms RANDOM algorithm in all three perspectives. Compared with counterparts, U2U-CF is biased as it produces significantly accurate but expected results.

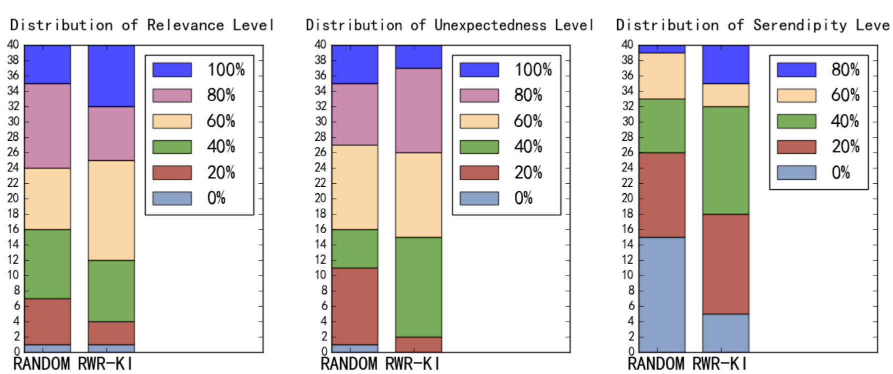
Figure 5(b) and Figure 5(c) presents the distribution of relevance, unexpectedness and serendipity level of across 5-item recommendation lists. It shows that 75% of the recommendation lists generated by CBR-FN algorithm contain at least three relevant items and 87.5% have more than three unexpected recommendations.



(a) Distribution of user ratings on recommendations.



(b) Distribution of serendipity, relevance and unexpectedness levels for CBR-FN and U2U-CF.



(c) Distribution of serendipity, relevance and unexpectedness levels for RANDOM and RWR-KI.

Figure 5: Online experiment result

4.Conclusion

In this project, we introduce the over-specialization problem and propose a content-based algorithm, named CBR-FN, that aims to add serendipity in recommendation results. An offline experiment on a user-movie rating dataset shows that the proposed CBR-FN algorithm shows better balance in relevance and unexpectedness than baseline methods. We have also conducted a real user study and proves that CBR-FN algorithm outperforms the other three algorithms in unexpectedness and serendipity and ranks second in relevance. Moreover, recommendations produced by CBR-FN algorithm are well accepted by users, since around 77% of the recommendations are relevant and 59% are serendipitous. Therefore, we have the evidence to believe that the CBR-FN can effectively help recommender systems to produce unexpected suggestions while narrowing the loss in relevance.

5. Future work

During the implementation and experimentation process, there are still some issues that need to be figured out in the future:

- The user-movie rating dataset contains some extreme unpopular movies (number of viewers less than 20), these unpopular movies might act as an negative factor to the performance and evaluation of the offline test.
- Scalability issue: at this stage, the online system takes around 2 seconds to generate recommendation list for each online user, might take some efforts to improve and optimize algorithms.
- During the experiment process, given a list of 20 ratings as the input, sometimes the system can produce similar recommendations to inputs with different values. One possible reason for this is that the database of the movie that we are using may be too small. Another reason may be that the 20 most popular movies from user's favorite genres still might not be enough to represent their interests.

Future work will aim at solving the above issues first, and then looking for ways to improve the algorithm. Details about our weekly logs can be found in the Appendix.

AppendixA. Weekly Logs

- Week 1 & 2: Apply Lemmatizing techniques to generate robust movie implicit features, look into Wikipedia dominance issue in knowledge infusion process and propose solutions. Conduct offline test using random algorithm and analyze results. Investigate Interest graph.

- Week 3 & 4: Implement proposed improvements to architectures of Knowledge Infusion process: 1. Parallel layer; 2. First layer Wikipedia, second layer Word Net; 3. First layer Word Net, second layer Wikipedia; Perform and compare results of different approaches. Come up with techniques to measure unexpectedness level in offline test.

- Week 5 & 6: Reorganize the offline experiment, study paper on serendipitous recommendation algorithms by Bhandari. Study Word2Vec model and related research papers, get hands-on practice on training this model.

- Week 7 & 8: Implement and train genism Word2Vec model with Wikipedia text source, implement feature network based on Word2Vec model, able to return shortest path between any two nodes in the graph. Adjust spreading network architecture.

- Week 9 & 10: Train LDA model of Wikipedia, construct feature network with genism Word2Vec model and Python package Networkx. Generate user and movie features based on the network.

- Week 11: Optimize video URL retrieving algorithm for the online movie system such that the intermediate loading time is decreased from 20 seconds to 2 seconds. Add a ranking page in the beginning.

- Week 12: Vectorize each user as a vector of length 10165. Further improve the design and integrate the result into recommendation procedures.

- Week 13: Improve feature network, upgrade Word2Vec model to support phrase similarity computation. Start to implement the collaborative filtering baseline method.

- Week 14: Finish movie and user profile generations, construct correlation matrix between user and movie implicit profiles. Recommend movie with the highest correlation value in the matrix.

- Week 15: Finish offline experiment and conduct online user study with masters from Amazon Mechanical Turk.

- Week 16: Finish up FYP report.

[1] Anderson, J. R. (1983). A spreading activation theory of memory. Journal of Verbal Learning and Verbal Behavior, 22, 261–295.

[2] Collins, A. M., & Loftus, E. F. (1975). A spreading activation theory of semantic processing. Psychological Review, 82, 407–428.

[3] Robertson, S. E., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 3, 333–389.

[4] Semeraro, G., Lops, P., Basile, P., & de Gemmis, M. (2009a). Knowledge infusion into content-based recommender systems. In Proceedings of the ACM conference on recommender systems, RecSys 2009 (pp. 301–304). ACM.

[5] Sparck-Jones, K., Walker, S., & Robertson, S. E. (2000). A probabilistic model of information retrieval: Development and comparative experiments – Part 1 and Part 2. Information Processing and Management, 36, 779–840.

[6] Gemmis, M. D., Lops, P., Semeraro, G., & Musto, C. (2015). An investigation on the serendipity problem in recommender systems.Information Processing & Management, 51(5), 695-717. doi:10.1016/j.ipm.2015.06.008

[7] Kotkov, Denis, Shuaiqiang Wang, and Jari Veijalainen. A Survey of Serendipity in Recommender Systems. Knowledge-Based Systems111 (2016). 180-92. Web.

[8] J. Corneli, A. Pease, S. Colton, A. Jordanous, C. Guckelsberger, Modelling serendipity in a computational context, Computing Research Repository abs/1411.0440.

[9] Toms, E. (2000). Serendipitous information retrieval. In Proceedings of DELOS workshop: Information seeking, searching and querying in digital libraries, ERCIM workshop proceedings – No. 01/W001

[10] Cantador, I., Brusilovsky, P., & Kuflik, T. (2011). 2nd Workshop on information heterogeneity and fusion in recommender systems (HetRec). In Proceedings of the ACM conference on recommender systems, RecSys 2011 (pp. 387–388). New York, NY, USA: ACM.

[11] Guanwen Yao, User-Based and Item-Based Collaborative Filtering. Recommendation Algorithms Design, 2015, UCSD, CS Department

[12] Tianming Wang, Introducing Serendipity in Recommender Systems, 2016, Final Year Project I

[13] Bhandari, U., Sugiyama, K., Datta, A., & Jindal, R. (2013). Serendipitous Recommendation for Mobile Apps Using Item-Item Similarity Graph. Information Retrieval Technology Lecture Notes in Computer Science, 440-451. doi:10.1007/978-3-642-45068-6_38

The Chinese University of Hong Kong
Academic Honesty Declaration Statement

Submission Details

Student Name	WANG, Tianming (1155029084)		
Year and Term	2016-2017 Term 2		
Course	IERG-4999-VJ01 Final Year Project II		
Assignment Marker	Professor Pascal Olivier VONTOBEL		
Submitted File Name	Introducing Serendipity in Recommender System.pdf		
Submission Type	Group		
Assignment Number	1	**Due Date (provided by student)**	2017-05-08
Submission Reference Number	1779671	**Submission Time**	2017-05-08 23:06:32

Agreement and Declaration on Student's Work Submitted to VeriGuide

VeriGuide is intended to help the University to assure that works submitted by students as part of course requirement are original, and that students receive the proper recognition and grades for doing so. The student, in submitting his/her work ("this Work") to VeriGuide, warrants that he/she is the lawful owner of the copyright of this Work. The student hereby grants a worldwide irrevocable non-exclusive perpetual licence in respect of the copyright in this Work to the University. The University will use this Work for the following purposes.

(a) Checking that this Work is original
The University needs to establish with reasonable confidence that this Work is original, before this Work can be marked or graded. For this purpose, VeriGuide will produce comparison reports showing any apparent similarities between this Work and other works, in order to provide data for teachers to decide, in the context of the particular subjects, course and assignment. However, any such reports that show the author's identity will only be made available to teachers, administrators and relevant committees in the University with a legitimate responsibility for marking, grading, examining, degree and other awards, quality assurance, and where necessary, for student discipline.

(b) Anonymous archive for reference in checking that future works submitted by other students of the University are original
The University will store this Work anonymously in an archive, to serve as one of the bases for comparison with future works submitted by other students of the University, in order to establish that the latter are original. For this purpose, every effort will be made to ensure this Work will be stored in a manner that would not reveal the author's identity, and that in exhibiting any comparison with other work, only relevant sentences/ parts of this Work with apparent similarities will be cited. In order to help the University to achieve anonymity, this Work submitted should not contain any reference to the student's name or identity except in designated places on the front page of this Work (which will allow this information to be removed before archival).

(c) Research and statistical reports
The University will also use the material for research on the methodology of textual comparisons and evaluations, on teaching and learning, and for the compilation of statistical reports. For this purpose, only the anonymously archived material will be used, so that student identity is not revealed.

I confirm that the above submission details are correct. I am submitting the assignment for:

[X] a group project on behalf of all members of the group. It is hereby confirmed that the submission is authorized by all members of the group, and all members of the group are required to sign this declaration.

We have read the above and in submitting this Work fully agree to all the terms. We declare that: (i) the assignment here submitted is original except for source material explicitly acknowledged; (ii) the piece of work, or a part of the piece of work has not been submitted for more than one purpose (e.g. to satisfy the requirements in two different courses) without declaration; and (iii) the submitted soft copy with details listed in the <Submission Details> is identical to the hard copy(ies), if any, which has(have) been / is(are) going to be submitted. We also acknowledge that we are aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the University website http://www.cuhk.edu.hk/policy/academichonesty/. We are aware that each student is responsible and liable to disciplinary actions should there be any plagiarized contents/undeclared multiple submission in the group project, irrespective of whether he/she has signed the declaration and whether he/she has contributed directly or indirectly to the problematic contents.

It is also understood that assignments without a properly signed declaration by the student concerned and by all members of the group concerned, will not be graded by the teacher(s).

Wang Tianming
Signature (WANG, Tianming, 1155029084)

2017-5-8
Date

Other Group Members (if any):

Name(s)	Student ID(s)	Signature(s)
HU JINWEN	1155028992	Hu Jinwen

Instruction for Submitting Hard Copy / Soft Copy of the Assignment

This signed declaration statement should be attached to the hard copy assignment or submission to the course teacher, according to the instructions as stipulated by the course teacher. If you are required to submit your assignment in soft copy only, please print out a copy of this signed declaration statement and hand it in separately to your course teacher.