International Journal of Computer Applications (0975 – 8887)
Volume 114 – No. 9, March 2015

Random Walk-based Recommendation with Restart using Social Information and Bayesian Transition Matrices

Suchit Pongnumkul
TMI, School of Engineering,
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan

Kazuyuki Motohashi
TMI, School of Engineering,
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan

ABSTRACT
A recommendation system is an information retrieval system that employs user, product, and other related information to infer relationships among data to offer product recommendations. The basic assumption is that friends or users with similar behavior will have similar interests. The large number of products available today makes it impossible for any user to explore all of them and increases the importance of recommendation systems. However, a recommendation system normally requires comprehensive data relating users and products. Insufficiently comprehensive data creates difficulties for creating good recommendations. Recommendation systems for incomplete data have become an active research area. One approach to solve this problem is to use random walk with restart (RWR), which significantly reduces the quantity of data required and has been shown to outperform collaborative filtering, the currently popular approach. This study explores how to increase the efficiency of the RWR approach. We replace transition matrices that use information regarding relationships between user, usage, and tags with transition matrices that use Bayesian probabilities, and we compare the efficiency of the two approaches using mean average precision. An experiment was conducted using music information data from last.fm. The result shows that our approach provides better recommendations.

General Terms
Recommendation system

Keywords
Recommendation system, random walk with restart, mean average precision

1. INTRODUCTION
Exponential increases in available information emphasize the importance of information filtering systems. As stated by Chris Anderson, "The secret to creating a thriving Long Tail business can be summarized in two imperatives: (1) Make everything available. (2) Help me find it." [1]. The success of any online service lies in helping users find what they are interested in, even before they realize that they are interested in it. Recommendation systems help online service providers offer personalized product suggestions by predicting user responses to items they have not yet considered.

Traditional recommendation systems achieve this goal by using either content-based filtering (CBF), which analyzes product characteristics, or collaborative filtering (CF), which analyzes user behaviors. However, the popularity of social networking has prompted researchers to use the concept of friendship and social information to increase recommendation accuracy. Konstas et al. [2] showed that a generic framework of random walk with restart (RWR) that includes social annotation (tags) and friendship established among users outperforms the currently popular CF approach. While social annotations and friendship data capture some information regarding how a user is related to a product, it does not provide a straightforward probability. The authors hypothesize that this relationship is captured more effectively using Bayesian probability. In this study, we extend the work of Konstas et al. [2] using Bayesian probability in RWR's transition matrices to increase recommendation accuracy.

We evaluated our modified model against the method proposed by Konstas et al. [2] on a data set collected from last.fm, an online music recommendation service. The dataset includes user, friendship, artist, and usage data. The modified model achieved better recommendation accuracy, particularly with limited data (80% of data removed). The contributions of this study include the following:

- We evaluatedd the use of Bayesian probability in RWR transition matrices and found that it outperforms the original RWR model.
- We changedd from track to artist recommendations.
- We found that changes in parameters (alpha) have little effect on recommendation accuracies.

The rest of this paper is organized as follows. Section 2 reviews previous related work. Section 3 describes the proposed method. The experiments, including how we collected data from last.fm, are explained in Section 4. We discuss the implications of our study and draw conclusions in Section 5 and Section 6, respectively.

2. RELATED WORK
2.1 Information Retrieval
Information retrieval (IR) is the finding of information that is relevant and satisfies an information need from a normally unstructured and large collection of information resources. As collection of information resources is big, people have difficulties in navigating within the collection manually. Such difficulties are called *information reload*, and automated information retrieval systems help reduce the difficulties [3].

The two most frequent and basic measures for information retrieval effectiveness are precision and recall [4]. Precision (P) is the percentage of retrieved information that are relevant and recall (R) is the percentage of relevant information that are retrieved, as calculated by formula (1) and (2).

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} \tag{1}$$

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} \tag{2}$$

The standard measure in the Text Retrieval Conference (TREC http://trec.nist.gov/) community is the mean average precision (MAP), which provides a single-figure quality measure, across recall levels. In particular, among available evaluation measures, MAP has been shown to have effective discrimination and stability [5].

For each information query, the average precision (AP) is calculated by averaging the precision value of a set of documents after each relevant document is retrieved. MAP is the average AP for all queries for all the related documents.

In other words, if the set of relevant documents for an information need "" $qj \in Q$ is $d_1, \ldots, d_{m_j}$, and R_{jk} is the set of ranked retrieval results from the top result until you arrive at document d_k, then

$$\mathrm{MAP}(Q) = \frac{1}{|Q|} \sum_{j=1}^{Q} \frac{1}{m_j} \sum_{k=1}^{m_j} \mathrm{Pr}ecision(R_{jk}) \quad (3)$$

2.2 Recommendation Systems

Recommendation systems, sometimes also called "recommender systems," are an extensive class of Web applications that involve predicting user responses to options [6].

There has been substantial research on this topic showing that a recommendation system can help users with information retrieval ([7], [8], [9], [10], [11], [12]).

Such systems typically use one of three approaches.

1. CF [13]: This method employs user behavior information, such as ratings or usage of an item, to find similar users and try to predict a missing usage or rating. There have been many studies using systems with this approach. As examples,
   - Cho et al. [14] use data mining and decision trees on web usage, while
   - Kim et al. [15] create groups of people who have similar activities with items.
2. CBF: This method uses item similarities and recommends the item closest to the items used by a target user. As examples,
   - Baraglia and Silvestri [16] cluster the contents and use the results for recommendation, while
   - Han et al. [17] introduce an algorithm based on rules created from combinations of items selected together.
3. Hybrid recommendation systems [18]: This approach uses information regarding both users and items.

Recent rapid social network growth has created interest in using social information in recommendation systems. Studies have shown that people on the same network or having friendship relations share similarities ([19], [20], [21]). This shows that social information can be used to suggest directions for finding recommendations based on relationship and item usage information.

Tagging, in which user-generated keywords are attached to online contents, is also used in recommendation systems, for example, to identify items to be retrieved in the future ([22], [23], [24], [25], [26]). Studies have shown a relationship between friendship and tagging, or social bookmarks ([27], [28]).

A tagging system (folksonomy) model is often characterized by a tripartite graph with hyperedges. The three disjoint, finite sets of such a graph correspond to

1. a set of persons or users $u \in U$
2. a set of resources or objects $o \in O$ and
3. a set of annotations or tags $t \in T$

which are used by users U to annotate objects O. A very general model of folksonomies is defined by a set of annotations $F \subset U \times T \times O$ ([29], [30], [31], [32]).

Studies have shown that tags can be used as inputs to a recommendation system ([33], [34], [35], [36]).

However, since a recommendation system uses information as input, a system cannot provide suitable recommendations, when there is insufficient or sparse data.

Huang et al. [37] created social item-and-user graphs and used graph analysis on this problem.

2.3 RWR

Random walk, a series of random variables [38], was first introduced by Karl Pearson in 1905 [39] and has been used in many fields. Google™'s well-known PageRank is based on random walk [40].

Random walk on graphs is a series of random variables X_i where X_i is a connected vertex selection for each node of each step ([41], [42], [43]).

RWR is the random walk that has probability α of jumping to the starting point, as shown in equation (4).

$$p^{t+1} = (1-\alpha)Sp^{t} + \alpha q \quad (4)$$

where p^t and p^{t+1} are the probabilities of remaining at each node at steps t and $t+1$, respectively, S is the transition matrix, α is the restart ratio, and q is the probability of remaining at each node at the starting step.

When the start stage probabilities are set equal, as in (4), the probabilities at the stable stage of node y show the relationship between nodes x and y ([44], [45]).

$$q_i = \begin{cases} 1 & i = x \\ 0 & i \neq x \end{cases} \quad (5)$$

2.4 RWR-based Recommendation System

Studies on the use of RWR on related topics include the following.

- Clements et al. used RWR in information retrieval [46].
- Craswell et al. used random walks to create rankings of documents for a given query [47].
- Barnd modeled consumer behavior as random walks on a weighted association graph [47].
- Fouss et al. present a new perspective on characterizing the similarities among elements of a database [48].

Furthermore, Konstas et al. [2] show results indicating that RWR outperforms the standard CF method using the four transition matrices shown in Fig. 1. The main target of using

RWR is to deal with cold start problems, which will occur when there is insufficient information.

	User	Track
User	I	UTr
Track	TrU	TrTr

3	User	Track	Tag
User	I	UTr	UTg
Track	TrU	TrTr	TrTg
Tag	TgU	TgTr	TgTg

4	User	Track
User	UU	UTr
Track	TrU	TrTr

5	User	Track	Tag
User	UU	UTr	UTg
Track	TrU	TrTr	TrTg
Tag	TgU	TgTr	TgTg

Fig. 1 Four types of transition adjacency matrices for a recommendation system using RWR

However, there are no studies regarding the effect of data size, recommendation accuracy, or restart ratio.

2.5 Bayesian Probability

Bayes' theorem was developed by Thomas Bayes using conditional probability [50]. Bayesian probability is based on the principle of a relative universe with k disjoint events (1, …, k) and an event . The conditional probability that an event, A_i, occurs, given that another event, E, has already occurred can be calculated using the following formula.

$$P(A_i \mid E) = \frac{P(E \mid A_i)P(A_i)}{\sum_{j=1}^{k} P(E \mid A_j)P(A_j)} = \frac{P(E \mid A_i)P(A_i)}{P(E)} \quad (6)$$

3. METHODOLOGY

RWR is used in creating recommendations from user, item, and social information similar to the method used by Konstas et al. [2]. The authors modified the method by creating relationships for artists instead of tracks and user-artist tags instead of user-track tags, as shown in Fig. 2 and equations (7)–(10). This modification reduces the number of calculations required. The transition matrices were changed to equations (23)–(31), as shown in Appendix 7.2.

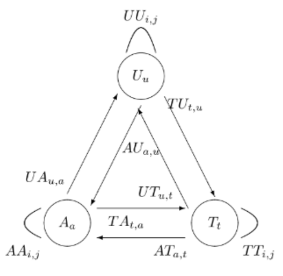

Fig 2 Graph for RWR

$$S_N = \begin{pmatrix} I & UA_{u,a} \\ AU_{a,u} & AA_{i,j} \end{pmatrix} \quad (7)$$

$$S_F = \begin{pmatrix} UU_{i,j} & UA_{u,a} \\ AU_{a,u} & AA_{i,j} \end{pmatrix} \quad (8)$$

$$S_T = \begin{pmatrix} I & UA_{u,a} & UT_{u,t} \\ AU_{a,u} & AA_{i,j} & AT_{a,t} \\ TU_{t,u} & TA_{t,a} & TT_{i,j} \end{pmatrix} \quad (9)$$

$$S_B = \begin{pmatrix} UU_{i,j} & UA_{u,a} & UT_{u,t} \\ AU_{a,u} & AA_{i,j} & AT_{a,t} \\ TU_{t,u} & TA_{t,a} & TT_{i,j} \end{pmatrix} \quad (10)$$

To improve recommendation accuracy, the traditional transition matrices are replaced with Bayesian probability transition matrices. The probability that node artist j is related to user i is calculated from the number of times that user i's friends listened to artist j divided by the number of times that everyone listened to artist j. This yields equations (14)–(22) in Appendix 7.1.

The constructed Bayesian transition matrices are used to perform RWR on the graph.

4. EXPERIMENTS

4.1 Data Collection

Data from last.fm were collected through their free web service API which provided data in xml format. The essential data includes user information, which can be obtained from user.getInfo. The user id list was not public information, but the authors worked around this problem by requesting random user ids and obtained more information from the user's friend relations (user.getFriends). Repeating this process provided us with a quantity of user information. Another essential datum is the artist information, which was obtained through user.getTopArtists. The information associated with this includes play counts (the number of times this user plays songs from this artist) and social tags (user.getTopTags). The relationships between users, artists, and tags were also collected, through user.getPersonalTags. The last.fm license agreement forbids obtaining an exhaustive list of data; hence, the authors collected a subset of data including 11,239 users, 49,000 artists, and 11,726 tags.

4.2 Data Set

The data from last.fm are too numerous to create an exhaustive transition matrix. To test our method, users and artists were randomly selected using the same method as that was used to collect the data. A user was first randomly selected and all the user's friends were included. Then, all the artists this user listened to and all the user's tags were included. This process was until the target data size was reached (Case 1: 400 users, 1,500 artists, and 600 tags; Case 2: 1,200 users, 3,000 artists, and 800 tags). For each case, some data regarding how users listened to their artists were randomly deleted by 20%, 50%, and 80%. Then the remaining data were used to create a transition matrix, as described in the Methodology section. For performance comparison, the authors also created the transition matrices, using the method described in [2]. This resulted in 12 data sets. Each data set received nine restart rations (α alphas) 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)

4.3 Evaluation

We evaluated our method by creating recommendations for each user using RWR for each data set and the 12 transition matrices. Each user has a sorted recommendation list based on artists' respective probabilities.

The RWR equation was solved $\hat{p}=aS\hat{p}+(1-a)q$ as

$$(I-aS)\hat{p}=(1-a)q \quad (11)$$

$$\hat{p}=(I-\alpha S)^{-1}(1-\alpha)q \qquad (12)$$

and the probabilities p and q were obtained at the convergence and restart stages.

Then, the artists who already exist in the user's history were removed and the authors compared the remaining artists to calculate precisions and MAPs.

$$MAP(Q)=\frac{1}{Q}\sum_{j=1}^{Q}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk}) \qquad (13)$$

where Precision(R_{jk}) = #artists who user j listened to in recommendation results from rank 1 to k, and m_j is the number of recommendations returned for each query (one query for each user).

5. RESULTS AND DISCUSSION

5.1 Comparing Effects of Restart Ratio

Fig. 3 shows the MAP effected by the restart ratio α. $\alpha = 0.4$ yields the highest MAP, but the change is insignificant.

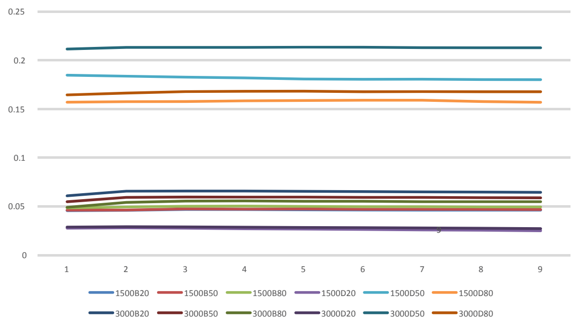

Fig 3 Comparison of the MAP of each result on change of α stepped by 0.1. The x-axis shows α × 10, and the y-axis shows the MAP. Twelve lines resulted from two sets of data size (1,500 and 3,000 artists) × 2 types of transition matrix (D for directed calculation and B for Bayesian-based transition matrices) × 3 sizes of listening data deletion (20%, 50%, and 80%). The result shows that α = 0.4 yields the maximum MAP.

5.2 Comparing Effects of Data Size

Fig. 4 shows the MAP for each test case group, with 20%, 50%, and 80% of the listening data deleted. This result shows that with the proposed method, the recommendation efficiency (MAP) is not affected by the quantity of data used.

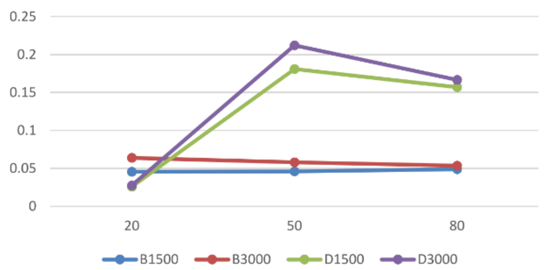

Fig. 4 Comparing the MAP of each result with change in data size. The x-axis shows change in data size, when the listening data were deleted 20%, 50%, and 80%, and the y-axis shows the MAP. The result shows that the proposed method performs better than previous methods for a small quantity of data.

5.3 Comparing effects of Transition Matrices

MAPs were not affected significantly by transition matrix type.

6. CONCLUSION

In the information overload era, recommendation systems are essential in helping users discover the information they are looking for. RWR has been used in recommendation systems to reduce the amount of information required to offer a recommendation. This study showed that RWR works well even in the case of limited data quantity. The effects of RWR's restart ratio was examined and the restart ratio α = 0.4 yielded the best recommendation accuracies. Bayesian probabilities was used in creating transition matrices and tested with a data set collected from last.fm. The results showed that Bayesian transition matrices outperform the traditional transition matrices.

Our method relies on RWR, whose creation and inversion of transition matrices require substantial memory. Therefore, for an increasing number of users or items, straightforward RWR implementation could result in memory or speed limitations. There has been research on increasing the efficiency of RWR that can be applied here to work around this problem.

Major directions for future work include improving the efficiency of using RWR in recommendation systems, particularly with regard to calculation costs. One approach is to select suitable iterative RWR methods or to compute the inverse of the transition matrix. The matrix inversion approach requires computing the inverse, which can be expensive, but once computed, the inverse can be readily used to compute recommendations. The iterative approach allows more frequent updates of the transition matrix. Another direction is to improve RWR scalability, because the sizes of social network graphs and items are constantly increasing. Yet another direction is to study how other social networking information (e.g., the number of messages sent or received between two friends, user viewing data, user following information, or friend suggestions) might improve recommendation accuracy.

Our proposed method is not limited to the music domain. RWR-based recommendation systems have been shown to work effectively in many domains ([51], [52], [53], [54], [55], [56], [57], [58], [59]), for which Bayesian transition matrices can also be introduced.

7. REFERENCES

[1] C. Anderson, The Long Tail: Why the Future of Business Is Selling Less of More, ser. Hyperion Books. Hyperion, 2006.

[2] I.Konstas,V.Stathopoulos,andJ.M.Jose,"Onsocialnetwork sandcollaborative recommendation," in Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ser. SIGIR '09. New York, NY, USA: ACM, pp. 195–202, 2009.

[3] Wikipedia. (2013, May) Information retrieval - wikipedia, the free encyclopedia. [Online; accessed 27-May-2013]. [Online]. Available:http://en.wikipedia.org/wiki/ Information_retrieval

[4] C. J. V. Rijsbergen, Information Retrieval, 2nd ed. Newton, MA, USA: Butterworth-Heinemann, 1979.

[5] C. D. Manning, P. Raghavan, and H. Sch¨utze, Introduction to Information Retrieval. New York, NY, USA: Cambridge University Press, 2008.

[6] A. Rajaraman and J. D. Ullman, Mining of Massive Datasets. New York, NY, USA: Cambridge University Press, 2011.

[7] F. Ricci, L. Rokach, B.Shapira, and P. B. Kantor, Recommender Systems Handbook, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[8] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, Recommender Systems: An Introduction, 1st ed. New York, NY, USA: Cambridge University Press, 2010.

[9] T. Hennig-Thurau, A. Marchand, and P. Marx, "Can automated group recommender systems help consumers make better choices?" Journal of Marketing, vol. 76, no. 5, pp. 89–109, 2012.

[10] R. Typke, F. Wiering, and R. C. Veltkamp, "A survey of music information retrieval systems," in IN ISMIR, pp. 153–160, 2005.

[11] A. Ansari, S. Essegaier, and R. Kohli, "Internet recommendation systems," Journal of Marketing Research, vol. 37, no. 3, pp. 363–375, 2000.

[12] F. M. Harper, X. Li, Y. Chen, and J. A. Konstan, "An economic model of user rating in an online recommender system," in Proceedings of the 10th international conference on User Modeling, ser. UM'05. Berlin, Heidelberg: Springer-Verlag, pp. 307–316, 2005.

[13] X. Su and T. M. Khoshgoftaar, "A survey of CF techniques," Advances in Artificial Intelligence, vol. 2009, pp. 4:2–4:2, 2009.

[14] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on web usage mining and decision tree induction," Expert Systems with Applications, vol. 23, no. 3, pp. 329–342, 2002.

[15] J. K. Kim, H. K. Kim, H. Y. Oh, and Y. U. Ryu, "A group recommendation system for online communities," International Journal of Information Management, vol. 30, no. 3, pp. 212–219, 2010.

[16] R. Baraglia and F. Silvestri, "An online recommender system for large web sites," in Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on, 2004, pp. 199–205.

[17] E.-H. S. Han and G. Karypis, "Feature-based recommendation system," in Proceedings of the 14th ACM international conference on Information and knowledge management, ser. CIKM '05. New York, NY, USA: ACM, pp. 446–452, 2005.

[18] R. Burke, "Hybrid recommender systems: Survey and experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.

[19] M. de Klepper, E. Sleebos, G. van de Bunt, and F. Agneessens, "Similarity in friendship networks: Selection or influence? the effect of constraining contexts and non-visible individual attributes," Social Networks, vol. 32, no. 1, pp. 82–90, 2010.

[20] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri,"Feedback effects between similarity and social influence in online communities," in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ser. KDD '08. New York, NY, USA: ACM, pp. 160–168, 2008.

[21] C. C. Aggarwal, Social Network Data Analytics, 1st ed. Springer Publishing Company, Incorporated, 2011.

[22] I. Peters, Folksonomies. Indexing and Retrieval in Web 2.0, 1st ed. Hawthorne, NJ, USA: Walter de Gruyter & Co., 2009.

[23] M. riitta Koivunen, "Semantic authoring by tagging with annotea social bookmarks and topics," in In The 5th International Semantic Web Conference (ISWC2006) - 1st Semantic Authoring and Annotation Workshop (SAAW2006, 2006).

[24] A. Hotho, R. J¨aschke, C. Schmitz, and G. Stumme, "Information retrieval in folksonomies: Search and ranking," in The Semantic Web: Research and Applications, ser. Lecture Notes in Computer Science, Y. Sure and J. Domingue, Eds. Springer Berlin Heidelberg, vol. 4011, pp. 411–426, 2006.

[25] S. Hayman, "Folksonomies and tagging: New developments in social bookmarking," in Ark Group Conference: Developing and Improving Classification Schemes. Citeseer, 2007.

[26] J. Trant, "Studying social tagging and folksonomy: A review and framework," Journal of Digital Information, vol. 10, no. 1, 2009.

[27] R. Angelova, M. Lipczak, E. Milios, and P. Pralat, "Investigating the properties of a social bookmarking and tagging network," International Journal of Data Warehousing and Mining, vol. 6, no. 1, pp. 1–19, 2010.

[28] P. Bhattacharyya, A. Garg, and S. Wu, "Analysis of user keyword similarity in online social networks," Social Network Analysis and Mining, vol. 1, no. 3, pp. 143–158, 2011.

[29] M. Strohmaier, "Purpose tagging: capturing user intent to assist goal oriented social search," in Proceedings of the 2008 ACM workshop on Search in social media, ser. SSM '08. New York, NY, USA: ACM, pp. 35–42, 2008.

[30] R. Lambiotte and M. Ausloos, "Collaborative tagging as a tripartite network," in Computational Science ICCS 2006, ser. Lecture Notes in Computer Science, V. Alexandrov, G. Albada, P. Sloot, and J. Dongarra, Eds. Springer Berlin Heidelberg, vol. 3993, pp. 1114–1117, 2006.

[31] P. Mika, "Ontologies are us: A unified model of social networks and semantics," in The Semantic Web ISWC 2005, ser. Lecture Notes in Computer Science, Y. Gil, E. Motta, V. Benjamins, and M. Musen, Eds. Springer Berlin Heidelberg, vol. 3729, pp. 522–536, 2005.

[32] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Can social bookmarking improve web search?" in Proceedings of the 2008 International Conference on Web Search and Data Mining, ser. WSDM '08. New York, NY, USA: ACM, pp. 195–206, 2008.

[33] L. B. Marinho, A. Hotho, R. Jschke, A. Nanopoulos, S. Rendle, L. Schmidt-Thieme, G. Stumme, and P. Symeonidis, Recommender Systems for Social Tagging Systems. Springer Publishing Company, Incorporated, 2012.

[34] F. Durao and P. Dolog, “A personalized tag-based recommendation in social web systems,” Adaptation and Personalization for Web, vol. 2, p. 40, 2009.

[35] K. Bischoff, C. S. firan, W. Nejdl, and R. Paiu, “Can all tags be used for search?” in Proceedings of the 17th ACM conference on Information and knowledge management, ser. CIKM ’08. New York, NY, USA: ACM, pp. 193–202, 2008.

[36] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel, “Social media recommendation based on people and tags,” in Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, ser. SIGIR ’10. New York, NY, USA: ACM, pp. 194–201, 2010.

[37] Z. Huang, D. Zeng, and H. Chen, “A link analysis approach to recommendation under sparse data,” in Proc. 2004 Americas Conf. Information Systems, 2004.

[38] C. M. Grinstead and J. L. Snell, Introduction to probability. American Mathematical Soc., 1998.

[39] K. Pearson, “The problem of the random walk,” Nature, vol. 72, no. 1865, pp. 294, 1905.

[40] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.” Stanford InfoLab, Technical Report 1999-66, November 1999, previous number = SIDL-WP-1999-0120.

[41] D. Aldous and J. fill, “Reversible markov chains and random walks on graphs,” 2002.

[42] L. Lov´asz, “Random walks on graphs: A survey,” Combinatorics, Paul erdos is eighty, vol. 2, no. 1, pp. 1–46, 1993.

[43] L. Lov´asz, “Random walks on graphs: A survey,” in Combinatorics, Paul Erd˝os is Eighty, D. Mikl´os, V. T. S´os, and T. Sz˝onyi, Eds. Budapest: J´anos Bolyai Mathematical Society, vol. 2, pp. 353–398, 1996.

[44] J. Xia, D. Caragea, and W. Hsu, “Bi-relational network analysis using a fast random walk with restart,” in Data Mining, 2009. ICDM ’09. Ninth IEEE International Conference on, 2009, pp. 1052–1057.

[45] H. Tong, C. Faloutsos, and J.-Y. Pan, “Random walk with restart: fast solutions and applications,” Knowl. Inf. Syst., vol. 14, no. 3, pp. 327–346, 2008.

[46] M. Clements, A. P. de Vries, and M. J. Reinders, “Optimizing single term queries using a personalized markov random walk over the social graph,” in Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR), 2008.

[47] N. Craswell and M. Szummer, “Random walks on the click graph,” in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ser. SIGIR ’07. New York, NY, USA: ACM, pp. 239–246, 2007.

[48] M. Brand, “A random walks perspective on maximizing satisfaction and profit,” in SIAM International Conference on Data Mining, pp. 12–19, 2005.

[49] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, “Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation,” IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 3, pp. 355–369, 2007.

[50] M. Nakatsuji, Y. Fujiwara, A. Tanaka, T. Uchiyama, and T. Ishida, “Recommendations over domain specific user graphs,” in Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence. Amsterdam, The Netherlands, The Netherlands: IOS Press, pp. 607–612, 2010.

[51] F. D´ıez, J. E. Chavarriaga, P. G. Campos, and A. Bellog´ın, “Movie recommendations based in explicit and implicit features extracted from the filmtipset dataset,” in Proceedings of the Workshop on Context Aware Movie Recommendation, ser. CAMRa ’10. New York, NY, USA: ACM, pp. 45–52, 2010.

[52] Z. Wang, Y. Tan, and M. Zhang, “Graph-based recommendation on social networks,” in Web Conference (APWEB), 2010 12th International Asia-Pacific, pp. 116–122, 2010.

[53] Y. Shi, M. Larson, and A. Hanjalic, “Mining relational context-aware graph for rater identification,” in Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation, ser. CAMRa ’11. New York, NY, USA: ACM, pp. 53–59, 2011.

[54] B. Liu and Z. Yuan, “Incorporating social networks and user opinions for collaborative recommendation: local trust network based method,” in Proceedings of the Workshop on Context-Aware Movie Recommendation, ser. CAMRa ’10. New York, NY, USA: ACM, pp. 53–56, 2010.

[55] Z. Wang, M. Zhang, Y. Tan, W. Wang, Y. Zhang, and L. Chen, “Recommendation algorithm based on graph-model considering user background information,” in Creating, Connecting and Collaborating through Computing (C5), 2011 Ninth International Conference on, pp. 32–39, 2011.

[56] J. Tang, S. Wu, J. Sun, and H. Su, “Cross-domain collaboration recommendation,” in Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ser. KDD ’12. New York, NY, USA: ACM, pp. 1285–1293, 2012.

[57] C. Lucchese, R. Perego, F. Silvestri, H. Vahabi, and R. Venturini, “How random walks can help tourism,” in Advances in Information Retrieval, ser. Lecture Notes in Computer Science, R. Baeza-Yates, A. Vries, H. Zaragoza, B. Cambazoglu, V. Murdock, R. Lempel, and F. Silvestri, Eds. Springer Berlin Heidelberg, vol. 7224, pp. 195–206, 2012.

[58] Y. Fujiwara, M. Nakatsuji, M. Onizuka, and M. Kitsuregawa, “Fast and exact top-k search for random walk with restart,” Proc. VLDB Endow., vol. 5, no. 5, pp. 442–453, 2012.

[59] M. Bayes and M. Price, “An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s.” Philosophical Transactions, vol. 53, pp. 370–418, 1763.

8. APPENDIX

8.1 Transition Matrices for Artist Recommendation USING RWR

$$UU_{i,j} = \frac{F_{i,j}}{\sum_{k=1}^{\#u} F_{k,j}} \tag{14}$$

$$UA_{u,a} = \frac{P_{u,a}}{\sum_{k=1}^{\#u} P_{k,a}} \tag{15}$$

$$UT_{u,t} = \frac{\sum_{k=1}^{\#a} T_{u,k,t}}{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} T_{i,j,t}} \tag{16}$$

$$AU_{a,u} = \frac{P_{u,a}}{\sum_{k=1}^{\#a} P_{u,k}} \tag{17}$$

$$AA_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{18}$$

$$AT_{a,t} = \frac{\sum_{k=1}^{\#u} T_{k,a,t}}{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} T_{i,j,t}} \tag{19}$$

$$TU_{t,u} = \frac{\sum_{k=1}^{\#a} T_{u,k,t}}{\sum_{i=1}^{\#a} \sum_{j=1}^{\#t} T_{u,i,j}} \tag{20}$$

$$TA_{t,a} = \frac{\sum_{k=1}^{\#u} T_{k,a,t}}{\sum_{i=1}^{\#u} \sum_{j=1}^{\#t} T_{i,a,j}} \tag{21}$$

$$TT_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{22}$$

where

#u = Number of users

#a = Number of artists

#t = Number of tags

$$F_{i,j} = \begin{cases} 1 & \text{User i and user j are} \\ 0 & \text{friends otherwise} \end{cases}$$

Pu,a = Number of playcount by user u on artist a

Tu,a,t = Number of tag by user u on artist a with tag t

8.2 Transition matrices for artist recommendation using RWR with Bayesian-based transition matrices

$$UU_{i,j} = \frac{\sum_{k=1}^{\#u} F_{k,i} F_{k,j}}{\sum_{k=1}^{\#u} F_{k,j}} \tag{23}$$

$$UA_{u,a} = \frac{\sum_{k=1}^{\#u} F_{k,u} P_{k,a}}{\sum_{k=1}^{\#u} P_{k,a}} \tag{24}$$

$$UT_{u,t} = \frac{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} F_{i,u} T_{i,j,t}}{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} T_{i,j,t}} \tag{25}$$

$$AU_{a,u} = \frac{\frac{\sum_{k=1}^{\#u} F_{k,u} P_{k,a}}{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} P_{i,j}}}{\frac{\sum_{k=1}^{\#u} F_{k,u}}{\#u}} \tag{26}$$

$$AA_{i,j} = \frac{\sum_{k=1}^{\#u} L_{k,i} P_{k,j}}{\sum_{k=1}^{\#u} P_{k,j}} \tag{27}$$

$$AT_{a,t} = \frac{\frac{\sum_{k=1}^{\#u} T_{k,a,t} P_{k,a}}{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} T_{i,j,t} P_{i,j}}}{\frac{\sum_{k=1}^{\#u} P_{k,a}}{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} P_{i,j}}} \tag{28}$$

$$TU_{t,u} = \frac{\frac{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} F_{i,u} T_{i,j,t}}{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} \sum_{k=1}^{\#t} T_{i,j,k}}}{\frac{\sum_{k=1}^{\#u} F_{k,u}}{\#u}} \tag{29}$$

$$TA_{t,a} = \frac{\frac{\sum_{k=1}^{\#u} T_{k,a,t} P_{k,a}}{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} T_{i,j,t} P_{i,j}}}{\frac{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} T_{i,j,t}}{\sum_{i=1}^{\#u} \sum_{j=1}^{\#a} \sum_{k=1}^{\#t} T_{i,j,k}}} \tag{30}$$

$$TT_{i,j} = \frac{\sum_{x=1}^{\#u} \sum_{y=1}^{\#a} W_{x,y,i} T_{x,y,j}}{\sum_{x=1}^{\#u} \sum_{y=1}^{\#a} \sum T_{x,y,j}} \tag{31}$$

where,

$$L_{u,a} = \begin{cases} 1 & \text{Pu,a > 0} \\ 0 & \text{Pu,a = 0} \end{cases}$$

$$W_{u,a,t} = \begin{cases} 1 & \text{Tu,a,t > 0} \\ 0 & \text{Tu,a,t = 0} \end{cases}$$