Incorpurating Social Relations into a Context Aware Music Recommendation System

Alieyeh Sarabandi, Terme Tabatabai, Natasha Ghaemi

Abstract—Social relations greatly influence users' musical preferences, so they are beneficial to music recommendation tasks. In this paper, we propose a simple approach to include social relations in an existing context-aware music recommendation approach. We attempted to improve this music recommendation approach by limiting the global music set to those pieces which have some form of personal connection (direct or indirect) to the user, therefore increasing the probability of successful recommendations. We do this by making use of users' social relations graphs to find music pieces which the user might like or be curious about. Experimental evaluations on a real-world dataset show that the proposed approach outperforms baseline methods in terms of precision, and hitrate.

Keywords—Recommender systems · Music recommendation · Context-aware recommendation · Social Relationships

1 Introduction

NOWADAYS, there is a huge amount of musical contents available on the Internet to choose from. This amount increases on a daily basis, while the big streaming services each offering billions of songs. Therefore, it becomes increasingly difficult for people to find music which suits their taste and musical needs. Recommender systems have emerged as a result, their purpose being to reduce the search costs and offer only the relevant items from enormous amounts of accessible data. Generally, traditional music recommender systems, such as collaborative filtering, content-based and hybrid approaches, try to solve the recommendation problem via the users' long-term music preferences. However, the context aware music recommendation method we based our work on takes into account contextual information such as physical surroundings, emotional state, time, weather, or the task at hand etc. The algorithm does so indirectly, by inferring contexts of listening to music from users' interactions with music systems. The users' historical listening records give us an idea of the users' preferences for music as well as other information while adding context can help recommend context appropriate music within the scope of the users' general preference. An important piece of information which greatly influences a uses' preferences is their social contacts. People are often influenced by their social circle and even if a particular piece isn't entirely in keeping with their prior preferences they tend to become curious if someone close to them or someone with a lot of influence on them likes a certain piece. People also tend to listen to the kind of music their favorite artists listen to. One thing to keep in mind is that while this information can be directly observed from the users' interaction with the system alone, we can also incorporate other social relations from other sources such as the users' contacts lists, who they follow on social media, etc. However we wont be getting into that in this paper. If the music system has a follow function we can use it as well as the time spent on each followees page to derive a social relations graph and each users relevance to the user we are recommending for. whether a follow function is not available or not, we can derive parts of, or all of this graph(depending on the presence of a follow function) by looking at the similarities between liked music and

- *A.Sarabandi is with the Department of Computer Engineering, Iran University of science and technology, Tehran, Iran,*
- *T.Tabatabai is with the Department of Computer Engineering, Iran University of science and technology, Tehran, Iran,*
- *N.Ghaemi is with the Department of Computer Engineering, Iran University of science and technology, Tehran, Iran,*

Manuscript received April 19, 2005; revised January 11, 2007.

preferences amongst users as well as incorporating the users' favorite artists into the mix if they also use that particular music system. We can use the aid of clustering methods for this. by doing so we get a better understanding of an important factor which influences users' music preferences. In this paper, we present an approach for incorporating social relations into an existing context-aware music recommendation system by limiting the global set of music for each user with relation to the music their accountancies and favorite artists like and listen to. In detail, our approach consists of three steps. Firstly, the proposed approach uses the users' social relations graph(which was in some way derived from the music system depending how said system functions) to find the most relevant users for each user we recommend to. Secondly, our approach uses these findings as well as the music preference of each user in this set to limit the users global music set. Finally, we give this set to the context aware music recommendation algorithm. Experimental evaluations show that the proposed approach has better performance than baseline approaches. Also we used to different algorithms for traversing our graph, one being a stack based algorithm and the other being a queue based algorithm. The result of our many experiments show that the two algorithms differ only slightly in speed depending on the sorting algorithm chosen. The stack based approach being faster for a user for whom the relevance of other users(present in his/her social relations graph) differs greatly, while the stack based approach is faster for a user for whom the relevance of other users is very close and similar. The remainder of this paper is structured as follows. Section 2 describes our problem statement. In Sections 3, we introduce the proposed approach in detail. Then, evaluations of the proposed approach are provided in Section 4. Finally, the conclusion and future work are given in Section 5.

2 Problem Statement

2.1 Preliminary Concepts

Definition 1. (Social Relation Graph): A social relations graph is a weighted tree whose nodes are users and its vertices are the relationship between users. Its root node is the user we want to recommend to. each nodes children are users who are immediately related to the node user. The weight of the vertices are the relevance that the child node have on its their parent node.

Definition 2. (Relevance): Relevance is the amount of presumed influence a user has on another. the higher the relevance of a user is to another the more likely it is for the first user to have an effect on the preferences or listening choices of the seconds user. It is an integer between 0 and 1, not counting zero itself.

Definition 3. (a context-aware music recommendation approach): Contextual factors greatly influence users' musical preferences, so they are beneficial to music recommendation and retrieval tasks. context aware recommendations utilize the contextual information and recommend accordingly. a context-aware music recommendation approach, can recommend music pieces appropriate for users' contextual preferences for music.

Definition 4. (acquaintance): In this paper when we use acquaintance we don't mean to say that the users are truly acquainted but that one of the nodes(users) is a ancestor of another in a social relations graph of a particular user. A direct acquaintance is in fact a child node while an indirect acquaintance is any acquaintance that isn't a direct acquaintance.

U	users
u_i	a user
V_i	music preference sorted
v_{ij}	j th music in V_i
R_{ij}	user j's relevance to user i
M	all music
Ms_i	u_i ' s global music list
f(I)=	$e^I R_{ij}$ (1)
I	importance of v_{ij} to u_i

2.2 Problem definition

given the music v_{ij} in lists V_i in for relevant users to u_i in U we want to pick the most important pieces(with highest f (I)s) to u_i and fill vertex $Ms_i \subseteq m$ them"

3 Proposed Approach in Detail

Our work is built upon 2 observations.

Observation 1: A lot of the time, users are more likely to listen to something an indirect acquaintance(node in lower layers of social relations graph) that is connected to them via an important direct acquaintance (high relevance) listens to than something an unimportant direct acquaintance listens to.

Observation 2: Users' preferences hardly ever deviate from the preferences of their connections on social media (nodes in social relations graph).

Based on the two observations mentioned above, we need a model that is capable of (1) picking the users that have the most influence on the usr we are recommending to, (2) picking music pieces which the user might like from amongst the music these users like, and (3) setting them as the users global music set.our model does exactly that with two slightly different algorithms.

3.1 Brief overview of our algorithm

Our algorithm takes the social relevance graph G of the user we are recommending to u_i and traverses it down to three layers (can be modified to more but using less layers is not recommended). The result is a set of users which will be stored in the vertex U_i which holds all relevant users to u_i down to three levels. We then sort this set from the most relevant user, to the least relevant user to u_i. After that we keep the first n users and remove the rest from U_i. Each user u_j in the set U_i has a set of music pieces M_j that he/she likes most and/or listens to most that is already sorted from most liked to least liked(how much it is like is a rational number between 1 to 3, we call this number I); The piece with the larger I is more well liked by u_j than a piece with a smaller index.

we use this I as the I in the equation (1) and we use the relevance between u_i and u_j as R_{ij} if they are direct acquaintances. However if u_i and u_j aren't direct acquaintances will have to calculate R_{ij}.

Imagine if p_{ij} is the shortest path between two(with length longer than 1) indirect acquaintances i and j with any node u_k in p_{ij}. R_{ij} will be

$$R_{ik} * R_{kj} \tag{2}$$

Now we proceed to calculating the results of (1) for each music piece m_i which is part of M_j, j being any user u_j in U_i. we do this for all users in U_i. Afterwards we but all these music pieces in to a vertex Ms_i. We now sort the pieces in Ms_i according to the results of equation (1) and discard the lesser half of the vertex leaving only the more relevant pieces.

we then use this as u_is global music list M and give it as an entry along with other necessary information to the context aware music recommendation algorithm.

3.2 Difference between our algorithms

The main difference between our algorithms is in the graph traversal section. We know that how sorted an array is before we run a sorting algorithm on it can have a major effect in the time complexity of quite a few sorting algorithms. However for our algorithms this difference isn't significant if the difference in relevance amongst vertices is small but for more sparse relevance we can see a somewhat noticeable difference.

3.3 Queue based algorithm

This algorithm uses Breadth-first search(BFS) as its graph traversal algorithm. The result goes in to the vertex U_i.

3.4 Stack based algorithm

Here we use a modified version of Depth-first search(DFS) for graph traversal. We take the node with the greatest relevance first. Then we take its most relevant child node. We continue this down to three layers(you can choose to go deeper) then we go back and pick the node with second highest relevance and we continue until we have traversed either all the nodes or a certain amount of nodes(we can choose not to bother with all the nodes and only choose to traverse until we have a certain amount of nodes) the result would be in the vertex U_i.

Algorithm 1 Queue based algorithm

procedure QUEUEBASED(G, s) ▷ Where G is the graph and s is the source node let Q be queue and res be the result queue

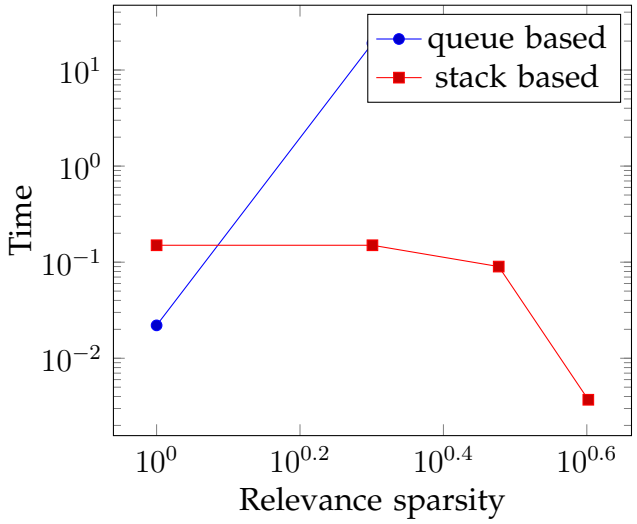
Q.enqueue(s) ▷ Inserting s in queue until all its neighbour vertices are marked.
$s \leftarrow visited$
while Q is not empty **do** ▷ Removing that vertex from queue,whose neighbour will be visited now $v \leftarrow$ Q.dequeue( )
while depth¡4 **do** ▷ processing all the neighbours of v for all neighbours w of v in Graph G
if w is not visited **then**
Q.enqueue(w)
$w \leftarrow visited$
res.enqueue(w)
end if
end while
end while
end procedure
function GETMUSIC(res, n)
List list ▷ list of favorite musics of top n users
res.sort(comparing(relavance))
for $i = 1$ to n **do**
res.get(i).favoriteMusics.sort(comparing rate)
for all $m \in res.get(i).M$ **do**
calculate(M) ▷ calculates f for each music in the favorite music list with the (relevance* (pow(e, rate))) formula
list.add(favoriteMusics)
end for
end for
list.sort() ▷ by comparing the f value of each music
return list
end function
function CALLRECOMMEND(res, n)
Recommend(list,s)
end function

Algorithm 2 Stack based algorithm

procedure STACKBASED(G, s) $s \leftarrow$ visited ▷ mark s as visited
for all neighbours w of s in Graph G **do** if(depht<4)
if w is not visited **then**
StackBased(G, w)
end if
end for
end procedure
function GETMUSIC(res, n)
List list ▷ list of favorite musics of top n users
res.sort(comparing(relavance))
for $i = 1$ to n **do**
res.get(i).favoriteMusics.sort(comparing rate)
for all $m \in res.get(i).M$ **do**
calculate(M) ▷ calculates f for each music in the favorite music list with the (relevance* (pow(e, rate))) formula
list.add(favoriteMusics)
end for
end for
list.sort() ▷ by comparing the f value of each music
return list
end function
function CALLRECOMMEND(res, n)
Recommend(list,s)
end function

4 EVALUATION OF PROPOSED APPROACH

In this section, we experimentally evaluate the performance of the algorithms as well as the proposed method.



As you can see we ran the two algorithms on 4 different datasets and measured the time. In these datasets the relevance of users in the social communications graph are the differing factor. The first set being closest to each other while the latest being(the one with the biggest sparsity number) has the most differing relevance amongst acquaintances. We can draw the conclusion that our stack based algorithm gives quicker results for sparser distributions of relevance while our queue based approach is better for similar relevances with few variation.

5 Related works

In this section, we provide a brief review of existing research related to our work, including recommendation systems, context-aware recommendation systems.

5.1 context-aware recommendation systems

In 2012, a survey was conducted by Gottwald and Koch [1], on recommender systems in data libraries. The survey and the description of solutions show that a totally satisfying approach for recommending publications does not exist; hence, this field needs further investigations in order to create a recommendation system that can understand context and recommend relevant information.

Early work in context-aware recommender utilize contextual information for pre-processing or post-processing [2], [3]. Recent work has focused on integrating contextual information with the user-item relations [4], [5], [6].

Herlocker et al. [2] improve the recommendation accuracy by using contextual information about the user's task. They conclude that incorporating information about the user's task into recommendation can lead to better performance in certain applications. The problem with this method is that it only operates within a 2D USER x ITEM space. In [3], a reduction-based pre-filtering approach is proposed, which uses the user's prior item preferences to help match the current context for recommending items. This approach is performed in two steps. First, it filters out ratings that do not match the current context. Then, traditional two-dimension recommendation algorithms are conducted on the reduced dataset. The drawback of this approach is the sparsity problem due to filtering the data.

There is also plenty of works done in factorization. for instance, In [7], random decision trees are applied to partition the user-item-context matrix. Then submatrices are factorized for capturing user and item latent preferences. In [4], a regression-based latent factor model is proposed to incorporate features and past interactions. In [5], a context-aware factorization machine is provided to simultaneously take context into account to enhance predictions. Xiong et al. [6] utilize tensor factorization for time-aware recommendation. In [8], High Order Singular Value Decomposition (HOSVD) is applied to factorize user-item-context space. The problem with these methods is that they are not feasible for scenarios with only implicit feedback data. Shi et. al. [9] propose to directly optimize Mean Average Precision (MAP) for context-aware recommendation. However, this method fails to take both take into account both user and item contexts.

According to Haruna et al.[10], depending on the nature of the context, contextual information can be extracted explicitly [11], [12], implicitly [13], [14], or using a machine learning approach [15]. In an explicit approach, users are required to provide the required information that describes their interests usually through ratings or asking direct questions. The system must be familiar with the vocabularies used. Information can also be acquired dynamically

by building a user profile that captures the preferences of users automatically from the environment. An initial set of keywords is usually provided by the user to initiate the process [16]. The system then subsequently uses this information to identify the documents and services potentially fitting the user's interests, and appropriately presents them. Machine learning is also used to design an automatic recommender system [17]. The idea is to use a statistical or data mining approach to infer the user's contextual information by monitoring his/her activities with the system rather than asking him/her to provide a predefined set of keywords describing his/her preferences.
Researchers usually combine both the explicit and implicit approaches when extracting users' information. This is because combining the two methods help in utilizing the advantages and getting rid of the drawbacks of each approach. In addition, combining the two approaches increases the flexibility of the RS. On the other hand, researchers preferred using implicit data than the explicit, this is because even though the explicit data provide the most reliable source of information, explicit data extraction is difficult to perform as many users do not want to give source of information, explicit data extraction is difficult to perform as many users do not want to out their information and besides they might not be sure of their interest [18]. On the other hand machine learning is relatively new and has a lot of room for growth, research in this field is still on going and promising.

6 Conclusion and Future Works

This paper presents a novel approach for incorporating social relationships into a context-aware music recommendation, which can limit the users' global music set and recommend appropriate music pieces that are in accordance with the users' preferences. Specifically, the proposed approach consists of three steps. Firstly, it uses the users' social relations graph to find the most relevant users for each user we recommend to. Then, it uses these findings as well as the music preference of each user in this set to limit the users global music set. Finally, we give this set to the context aware music recommendation algorithm. Experimental evaluations on a real world dataset show that the proposed approach outperforms baseline methods.

Our work differs from prior works in one aspect, that being that it proposes two algorithms, which can be chosen based on how narrow a users relations are. Both of which being effective but naive enough to be extremely easy to understand and implement for almost any developer.

Based on our current work, there are three possible future directions. First, we can connect blog services (such as Tumblr) and google search histories with music service websites to improve our approach, and adopt more advanced techniques to further improve the performance. Secondly, this work focuses on deriving information from the users' social relations graph, we will explore the different ways we can derive the graph itself from music services, based on features they have in common. Finally, in real life the relevance of users to each other is dynamic, therefore we incorporate a method for keeping the social relations graphs up to date.

Alieyeh Sarabandi undergraduate student at Departments of computer engineering, Irans university of science and technology,Tehran, Iran.
email: Alieyeh@gmail.com

Terme Tabatabai undergraduate student at Departments of computer engineering, Irans university of science and technology,Tehran, Iran.
email: termehtabatabaee@yahoo.com

Natasha Ghaemi undergraduate student at Departments of computer engineering, Irans university of science and technology,Tehran, Iran.
email: Naatasha.gh@gmail.com

REFERENCES

[1] S. Gottwald and T. Koch, "Recommender systems for libraries," in *Proceedings of the ACM international conference on Recommender systems*, 2011, pp. 1–5.

[2] J. L. Herlocker and J. A. Konstan, "Content-independent task-focused recommendation," *IEEE Internet Computing*, no. 6, pp. 40–47, 2001.

[3] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems (TOIS)*, vol. 23, no. 1, pp. 103–145, 2005.

[4] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 19–28.

[5] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 635–644.

[6] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with bayesian probabilistic tensor factorization," in *Proceedings of the 2010 SIAM international conference on data mining*. SIAM, 2010, pp. 211–222.

[7] X. Liu and K. Aberer, "Soco: a social network aided context-aware recommender system," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 781–802.

[8] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 79–86.

[9] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver, "Tfmap: optimizing map for top-n context-aware recommendation," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 155–164.

[10] K. Haruna, M. Akmar Ismail, S. Suhendroyono, D. Damiasih, A. Pierewan, H. Chiroma, and T. Herawan, "Context-aware recommender system: a review of recent developmental process and future research direction," *Applied Sciences*, vol. 7, no. 12, p. 1211, 2017.

[11] W. Yuan, D. Guan, Y.-K. Lee, S. Lee, and S. J. Hur, "Improved trust-aware recommender system using small-worldness of trust networks," *Knowledge-Based Systems*, vol. 23, no. 3, pp. 232–238, 2010.

[12] W. Yao, J. He, G. Huang, J. Cao, and Y. Zhang, "A graph-based model for context-aware recommendation using implicit feedback data," *World wide web*, vol. 18, no. 5, pp. 1351–1371, 2015.

[13] E. R. Núñez-Valdéz, J. M. C. Lovelle, O. S. Martínez, V. García-Díaz, P. O. De Pablos, and C. E. M. Marín, "Implicit feedback techniques on recommender systems applied to electronic books," *Computers in Human Behavior*, vol. 28, no. 4, pp. 1186–1193, 2012.

[14] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*. ACM press New York, 1999, vol. 463.

[15] L. M. López-López, J. J. Castro-Schez, D. Vallejo-Fernandez, and J. Albusac, "A recommender system based on a machine learning algorithm for b2c portals," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1. IEEE, 2009, pp. 524–531.

[16] Y.-M. Li and C.-P. Kao, "Trepps: A trust-based recommender system for peer production services," *Expert systems with applications*, vol. 36, no. 2, pp. 3263–3277, 2009.

[17] G. Shani, L. Rokach, A. Meisles, L. Naamani, N. Piratla, and D. Ben-Shimon, "Establishing user profiles in the mediascout recommender system," in *2007 IEEE Symposium on Computational Intelligence and Data Mining*. IEEE, 2007, pp. 470–476.

[18] D. Bouneffouf, "Towards user profile modelling in recommender system," *arXiv preprint arXiv:1305.1114*, 2013.