Music Recommendation System Based on Last.fm Datasets

Haoyou Liu, Huanchen Lu, Junxu Lu, Muhan Yuan

1. Research Question and Motivation

Recommendation system is a hot topic in information system that seeks to predict the preference of users based on the pattern of user behavior. In this project, we are interested in which factors and algorithms can best predict users behavior. Also, we would like to visualize a music network based on user behavior.

During the project, we utilized the listening history of users from Last.fm to establish a music recommendation system. Specifically, we tried these methods to recommend artists to users: 1) Artists recommendation based on cosine similarity; 2) "Simplified" random walk on bipartite network; 3) Collaborative filtering. By comparing the accuracy, we can find a relatively good method to recommend artists to users.

2. Related work

Our related work is mainly divided in three parts: 1) Bipartite network projection and personal recommendation; 2) measure of structure-context similarity; 3) Integrated Diffusion on user and tag.

Music recommendation has attracted a lot of interests from the scientific domain since it has many real life applications and bears multiple challenges. The way to improve music recommendations has attracted equal attention. In [1], we built and evaluated a weighted Bipartite network projection that incorporate users and artists' relationship. Furthermore, the personal recommendation it elaborated also helped us propose a recommendation algorithm, which is a direct application of the weighting method for bipartite networks. Their study provides insights into the way to determine the edge weight in projection.

The next class of related work concerns measure of structure-context similarity. A detailed overview of methods can be found in [2]. In this study, it mentions about random surfer-pairs model, this model is present in the context of general direct graphs; variations for bipartite. Relevant to our study is the work which present the importance and the method for user and artist modeling, while implementing such evaluations simultaneously by the same user in different way. We simplified our model of random walk to two steps to avoid the computing-intensive, it could increase the efficiency of million songs dataset process. In addition, the author of [2], proposed a good overview of vector cosine-similarity and the pearson-correlation. This method help us compute the similarity between set of objects, like user and artist.

The last principle of related work [3] refers to personalized recommendation via integrated diffusion on user and tag, this paper only provides a simple start point for the design of hybrid algorithms making use

of tag information. In the similar way, we build the unweighted bipartite network projection based on tag, also the tag information can also be exploited under framework of collaborative filtering and diffusion algorithm. For the future work, we will try collaborative filtering and diffusion based on tag instead of just based on artist.

3. Data

The main source of datasets we used for this project was Last.fm Dataset - 360k users, released by the Universitat Pompeu Fabra on March 2010. It contains millions user listening history and metadata features for around 360000 unique users and 17.6 million records, varying in user_id, artist name and playcount. Supplementary dataset we collected from Last.fm API by ourselves and matched to artists in the Last.fm Dataset was also used. This includes artist_id and tags.

- ***User Listening History***

  This data comes in a tsv form of the artists a given user has listened to, with the playcount for certain artist. Throughout the project, we combine the two datasets, and we have roughly modeled user preferences by utilizing cosine similarity. Therefore the methods we implemented below for weight Bipartite network projection were based on this assumption.

- ***Artist Tags***

  This data comes in json format of the tags users gave to a certain artist, with more than 300k records. Due to the inconsistency of the tag labels the users gave, we decided that information like tag name, would be normalized before data processing and modeling.

4. Methods

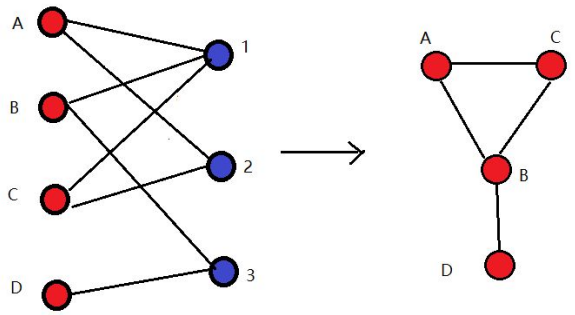
Bipartite

We modeled the relations between users and artists as a bipartite. There will be an edge between one user and one artist if the user has listened to this artist. The edge weight is determined by either the raw play count or the play count ratio (ratio = $\frac{how\ many\ times\ listen\ to\ this\ artist}{user's\ total\ play\ count}$).

Similarly, we draw the bipartite of artists and tags based on the tags one artist has. Different from the user-artist bipartite, since we cannot measure how close one artist is linked to one tag, this bipartite is unweighted.

One-mode Projection

To gain a more clear insight of the relations between artists, we did one-mode projection onto artist. In this network, two artists are connected if at least one user listens to both of them.



We tried three methods to measure the edge weight of this bipartite. Projecting it on an unweighted network is the most intuitive way, but it failed to show the real relation between artists. Since most artists share more or less users no matter how big the play count is, the network would be close to a complete graph. To solve this problem, we set a threshold for this projection: only users listen to one artist more than 10 percent of their total play count will be taken into consideration. In this case, two artists are connected only if they share a "big fan". And the weight is measured by the interaction of their frequent listeners over the union of their frequent listeners. To take a step further, instead of considering different users are same to the artist, we weighted different users based on the play count proportion.

Similarity Based Recommendation

Our final goal is to make recommendation for a user based on their listening history. To facilitate our following analysis, we represented the network as an adjacency matrix, from which we can find the similar artists of one artist by calculating the cosine similarity between each pair of vectors.

The most intuitive way to recommend a new artist to a user is find of artists that are similar with artists this user is currently listening to. Based on this idea, we generate a score for each artist using the user's listening history and the artist adjacency matrix.

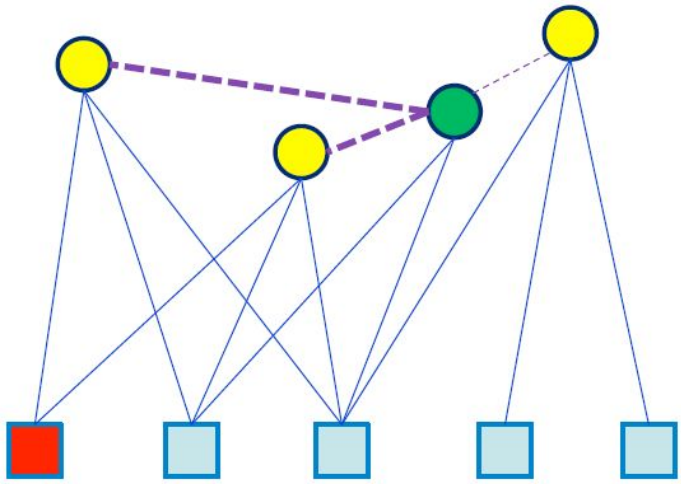
$$Score(j) = \sum_i Similarity(i,j) \times \frac{User\ play\ count\ of\ artist\ i}{User\ total\ play\ count}$$

Then, we can rank these score from highest to lowest and return top results as recommendation list.

Collaborative filtering

Collaborative filtering is a frequently used technique for recommender systems. The assumption of collaborative filtering is that users with a common interest will have similar preferences, so we could make predictions for an individual user based on the judgements of other similar users. The network

interpretation of collaborative filtering can be represented as:



Specifically, we used user play count ratio to establish user vectors and calculated the cosine similarity between each pair of users. After we obtained the similar user lists for all users, we used memory-based approach to make our predictions:

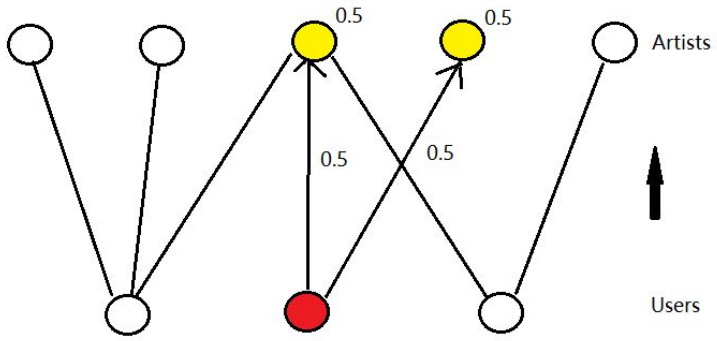
$$prediction(i,\ a)\ = \sum_{j} Similarity(i,j)\ \times\ (\ p(j,\ a)\ -\ n(j)\ )\ +\ n(i)$$

- Prediction(i, a): prediction score for user i's preference of artist a
- Similarity(i, j): the cosine similarity between user i and user j
- p(j, a): user j's preference of artist a
- n(i): user i's average preference of all artists
- Preference is measured by play count ratio

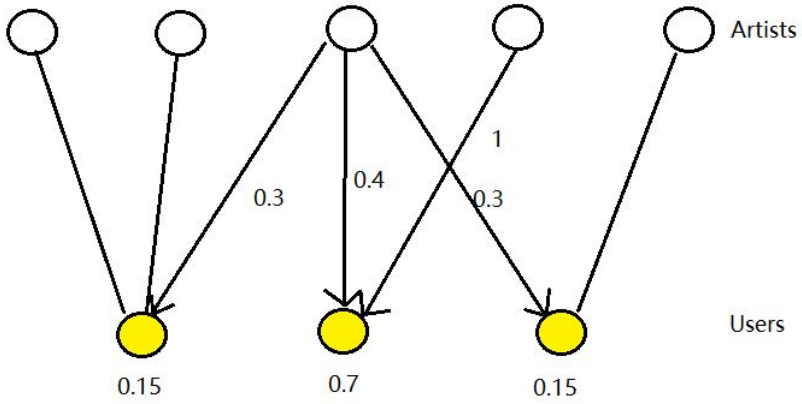
Resource-Allocation

In a traffic network, passengers can be considered as a find of resource that flow between different stations. In this user-artist network, users' preference can be perceived as resource that can be distributed to neighboring nodes.

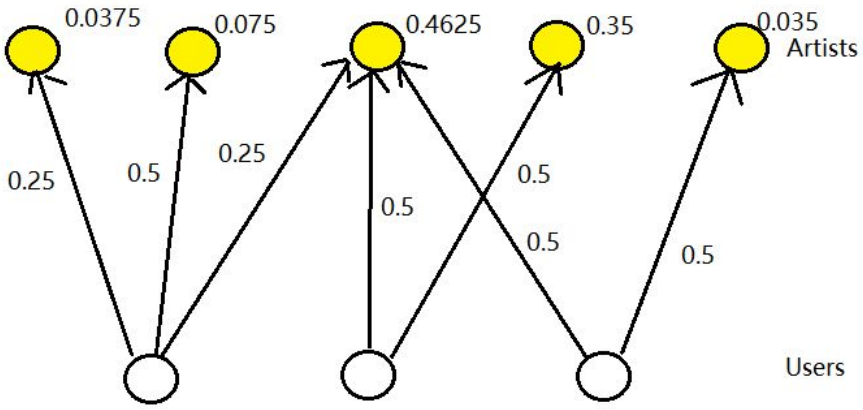
When it comes to making recommendation for one user, at time 0, all resources are located at artists that his has listened to in the past. The amount of resources on each artist is decided by the proportion of this user listens to the artist. The summation of resource at any time would be 1.



At time 1, all resources on artists are "diffused" to users and the proportion is determined by the how many play counts of the artist come from the user.



At time 2, again, all resources on users go back to artists based current resource of each user and his listening proportion.



The number of step needs to be even so that all resources would be on artist side.

Resource allocation would be an approximation of Random Walk after sufficiently large time n. However, in our project, we don't plan to go that far for two reasons. The first reason is a huge amount of calculation will be involved in the resource allocation simulation. It would be extremely time consuming to perform allocation that has more than 10 steps. And another reason is, we are aiming to provide a personalized recommendation for the user. The more steps resource allocation goes, the less personalized the recommendation result would be. So, we tried to find a balance point between recommending "good" artists and recommending artist similar with other artists the user listens to.

5. Evaluation and Results

Evaluation

We extracted one artist from each user's listening history to form our test dataset. We tested our methods by giving top 50 predictions for each user and see if the extracted artist is in the prediction list. Our baseline accuracy is 16.67% because we have 300 artists in total and by randomly choosing 50 artists from the artist list, we can achieve this baseline accuracy.

Cosine Similarity between artists

- Accuray: 62.76%

Surprisingly, the simplest method has the best result accuracy. We think the reason behind this is Cosine similarity method keeps the targeting user's feature to the maximum. However, we should realize that there could be some potential disadvantages for simple cosine similarity method. Since we set a threshold for cosine similarity calculation, some artist might have no link to artists the user listened in the past. That means these artists may never show up in the recommendation list, which indirectly sacrifices the diversity of the recommendation.

Collaborative filtering

- Accuracy: 42%

The collaborative filtering is a tentative methods for our project and memory-based filtering will take extremely large storage space. So, we randomly sampled 1,000 users for this test. And this may partly explains why the accuracy is relatively low.

Resource Allocation

- Accuracy:
    - 2 steps: 57%
    - 4 steps: 48%
    - 10 steps: 37%

For Resource Allocation, the accuracy is also not better than Cosine Similarity method. The more step it goes, the “worse” the result would be, since the recommendation become more generalized rather than personalized.

Although the result is not more precise comparing to cosine similarity, we should not simply consider this as a bad recommendation, because the philosophy of recommendation system is not simply predicting users’ preference, the quality, diversity of recommendation result are also important aspects.

6. Challenges

Throughout our project, there are many interesting problems to tackle when we process and model our dataset. One issue that makes the data analysis difficult is the huge size of the dataset, however, our task was somewhat huge database based: we are trying to predict what artists the user might like according to his or her listening history based on million datasets for analyze the similar artists. It tooks a long time for us to run the resource allocation model, thus, we need to figure out the way to simplify the algorithm and analyzing processing. Another issue is in identifying useful features of the users; determining which features would be best for a creating test and train dataset is important, since good feature selection could improve recommendations. Lastly, we decided to generate a score for each artist using the user's listening history and the artist adjacency matrix, to train our recommendation system, and the scope of this data presented issues.

7.Conclusions

Our work has given us a clear insight into recommendation systems for music, it show us the way how it works. We compared and tested the performance of the dataset by multiple methods: collaborative filtering, resource allocation and cosine similarity between artists. The procedure which performed the best was cosine similarity for which we obtained a precision of 62.76%.

The main challenge in getting these procedures to yield a satisfactory evaluation precision is that cosine similarity shows the best performance. However, our task was somewhat different: we are trying to predict what artists the user might like according to his or her listening history based on this million datasets, it spent a long time for us to process these million datasets by utilizing cosine similarity, let alone resource allocation. In future work, we would like to explore variants of the latent factor model, and in the meantime, try these models algorithm based on tags. Furthermore, we can optimize our resource allocation in a more mature and advanced way.

8.References

[1] Tao Zhou, Jie Ren, Matus Medo, Yi-Cheng Zhang: Bipartite network projection and personal recommendation.(2007)

[2] Glen Jeh, Jennifer Widom: SimRank: A Measure of Structural-Context Similarity.(2001)

[3] Zi-Ke Zhang, Tao Zhou and Yi-Cheng Zhang: Personalized Recommendation via Integrated Diffusion on User-Item- Tag Tripartite Graphs.

[4] Liu, Run-Ran, et al. "Personal recommendation via modified collaborative filtering." Physica A: Statistical Mechanics and its Applications 388.4 (2009): 462-468.

[5] Zhang, Zi-Ke, Tao Zhou, and Yi-Cheng Zhang. "Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs." Physica A: Statistical Mechanics and its Applications 389.1 (2010): 179-186.

[6] Ou, Qing, et al. "Power-law strength-degree correlation from resource-allocation dynamics on weighted networks." Physical Review E 75.2 (2007): 021102.

[7] Yu, Fei, et al. "Network-based recommendation algorithms: A review." *Physica A: Statistical Mechanics and its Applications* 452 (2016): 192-208.