Modeling and Analyzing User's Music Taste Shift with Amazon Reviews

Jack Mi

Abstract:

In this project, I aim to investigate how people's taste of music shift through time by analyzing Amazon Review data, focusing on the reviews in purchases around music (albums, digital albums, musical instruments). I think it is interesting to see how people's musical taste change in time, especially the taste related to genres. While there are multiple existing recommender systems that uses graphic models, the aim of this paper is not to try another method to predict the next purchase of the user, but instead more of a sociological one that tries to find the best model to depict Amazon users' typical shift of music taste, and tries to find a meaningful interpretation. However, graph-based recommender systems can be useful pointers and even metrics to decide if the experiment is successful or not.

1. Introduction:

People's music taste shifts in time frequently. For many people in their life span, they change their favorite genres all the time. Some people might tend to jump from genre to genre frequently: rock to emo, jazz to blues, etc. Others might stick to some particular genres, such as the hard-core metal fans and classical listeners. Thanks to this era of big data, we are now able to collect the history of a huge number of users online and investigate how their music taste shifts through time and attempt to conclude some patterns from their behavior. By understanding people's music taste, we might gain better understanding of the genre classification in the current music industry, and eventually sort, combine, and even create genres more scientifically in the future.

The main dataset I will be using is the Amazon review data in digital music section, collected by Julian McAuley from UCSD (McAuley, 2016). This dataset spans an 18-year time period, which is a decently long time for such time related task. In this paper, I will be building multiple graphic models on music genre incrementally, discuss their assumptions and intuitive limitations, and assess their expressivity in depicting the general trend of user's music taste shift. As metrics, standard role detection and link prediction methods will be used to assess how accurate each model depicts the music taste shift. The models go from simple to complex: from simply counting the consequent purchases (by counting ratings as an approximate) of digital music to build a genre flow graph, to considering the indirect influence of non-consequent purchases, and finally to utilizing timestamp information, rating information and user information.

2. Previous Works:

For previous works, I selected a set of diverse papers that are relevant to this problem. They are related to the question in different ways. Some are related to the traditional way of such analysis which might provide us some guidance on the several aspects of attention and motivation behind. Some are related to the graphical model aspects that I will be referring to. Some are related to the metric for assessment of each model.

The study of music taste has been a traditional topic in music history and musicology. Studying music taste shift can tell us a lot about sociological events and information about some culture. There are many rigorous academic papers on this field, one of them being "Modelling the Public's Taste: Local Habits, Ethnic Pluralism and European Music in Bucharest (1821–1862)" published in 2017 written by Haiganuş Preda-Schimek (Preda-Schimek, 2017). The author dived deep into historical and cultural roots together with the shift of political events to analyze the Ramanan public's music taste shift. Admittedly this is not a paper in the field of computer science, it still provides some insight about the aspects a traditional trained musicologist would consider when tackling a query about taste shift. Since the author's object is a traditional period in a foreign land, he is mainly dealing from a historical point of view. It is thrilling to try out a research using the power of big data and graphical models.

Similar study using similar data has been conducted in different fields. In the paper "From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise through Online Reviews" published in 2013 by J. McAuley and J. Leskovec is an excellent example demonstrating the usage of Amazon review data in analyzing and modeling user's behavior in a confined field: fine food (McAuley & Leskovec, 2013). In this paper, the authors use the Amazon review data about fine food and focused on a temporal analysis for users. In the paper, they experimented on using a sequence of reviews by the user and try to learn how the user's purchasing behavior and review rating develop through the process of purchasing different types of food. Their model is essentially graph-based, which provides much insight in methodology.

Other related fields are the online recommender systems, which has been a popular field and has been proven useful to large companies. As a representative, the paper "Pixie: A System for Recommending 3+ Billion Items to 200+ Million Users in Real-Time" (Eksombatchai, et al. 2018) provide a cutting-edge recommender system developed for Pinterest. It uses a variation of random walk algorithm to produce extremely accurate recommendations with little latency. It would be a very good metric for model assessment tool for my purpose. I can also use random walk algorithm to make link prediction, compare it to the true ratings, and assess how much information my models are capturing correctly.

3. Dataset:

As briefly mentioned in the introduction section, the main dataset I will be using is the Amazon review data in digital music section (rating only), collected by Julian McAuley from UCSD (McAuley, 2016). The dataset contains 836,006 ratings from Amazon users in digital music department. For each entry, it has the (user, item, rating, timestamp) tuples. Combined with the meta-data set collected by Jure Leskovec (Leskovec, 2007), I am able to map each item

to its genre and try to summarize the purchase history in the genre level. Admittedly, the dataset is still not comprehensive enough, as not all users purchase a decent number of times. Other possible dataset that might fit this analytical work might be YouTube music video playing history or Spotify music playing history. However, these datasets: 1, can be extremely large and might be very noisy. 2, does not have easy public access as the already collected Amazon dataset. Therefore, I decide to first experiment with this Amazon rating dataset and see how well the models work.

4. Building Models Incrementally

There are many possibilities when building statistic models to understand the public's music taste shift. The main idea is to build a model to represent the interaction between different genres. As a basic idea, we can portrait the flow of music taste shift between genre as a directed graph. For example, if a person tends to start by listening to only rock music, and then shift to become a classical music fan, there should be an edge from rock music to classical music. If a person tends to listen to both jazz and blues music, there should be a pair of reciprocal edges between jazz and blues.

It is best to construct the model incrementally and test it stepwise, since there is so much information that can be used or interpreted in different ways. As side notes, for all underlying models (that is all models except model 4, the complete model), if a song belongs to multiple genres, then we make a link between each pair of the genres. I also assume that if a user purchases a song or an album on Amazon, then it at least means that they have some interest in the kind of genre, even as an attempt. These attempts, while the user might not end up liking the genre, are a part of the musical genre interest flow, and are a part of the content that I try to capture in my models.

4.1. Model 1: Naïve Consequent Rating Instance Counting

In this model, we naively assume that purchase means interest, no matter its ratings. We also assume naively that every purchase is only affected by the immediate previous purchase. Finally, we assume that time does not have an effect on taste shift. Under these assumptions, we are able to come up with a naïve model from the data.

For construction, create a node for each genre. Sort the dataset in user. Create a linear rating history for each user. For each consequent two ratings for genre A and B, add a directed edge (A, B). After looping over all the data, count the number of edges between each two genres and assign it as the weight of the final edge (A, B). Note that we allow self-edges.

As a short summary, the resulting graph has high self-edge weights for every node, which indicates that people tend to stick with their own genres. Popular genres have high edge weights since more people rated these, and similar genres have high weight edges between them.

There are several obvious limitations in this model. Some of them are due to the specific model design which makes them fixable, some of them are due to assumptions which would require new models.

1, The created graph is useless for any person with zero purchase history. This can be solved easily by adding a starting node. For the first rating of any user to genre A, add an edge (Start, A).

2, If two items are purchased with the same timestamp, with any tie breaking tool we are only able to add one edge, which breaks the symmetry. A potential fix is to add two edges if two items are purchased simultaneously, but this will break cause simultaneous purchases to each create two edges instead of one edge, and also cause trouble when creating the next link: do we also create two edges for the purchase afterward? These problems will be fixed in a model that take timestamp into consideration.

3, The rating is not taken into consideration. If a person likes jazz (always rating 5 stars), then the person tries out rock and finds it terrible (rates it 1 star) and swears never to listen to rock again, it would be a mistake to make this consequent purchase a boost in weight of (Jazz, Rock). This problem will be fixed in a model that take rating into consideration.

4.2. Model 2: Timestamp Based Weight Assignments

In this model, we improve from the naïve model with the update of several assumptions: user's music taste is influenced by the entire history of the user's rating history, not only the previous one. However, the influence decrease as the time interval increase. The other assumptions remain the same (i.e. rating does not matter).

For construction, create a node for each genre. Create a node for starting point. Sort the dataset in user. Create a linear rating history for each user. For each two ratings for genre A and B where genre A rating happens before genre B, add a directed edge (A, B) with weight

$$w = \frac{1}{1 + \alpha\big(timestamp(B) - timestamp(A)\big)}$$

, where α is a parameter to control the tuning effect of time interval. After looping over all the data, sum up the edge weights between each two genres and assign it as the weight of the final edge (A, B).

Note that as long as the weight is inversely related to the time interval, we are able to depict the decreasing influence related to time interval. There are many ways to assign weight. In our case, notice the two special cases: when $\alpha = 0$, we consider each pair of rating equally important regardless of the time between them. When α is relatively large, we degenerate to counting only the simultaneous ratings. In general, α should be relatively small and should be controlled closely so we can study a reasonable range of time. For example, if we are more interested in the influence within one month, we should tune α so a one-month interval is in similar magnitude with 1. Other variations can be taking power root or taking power of the time

interval, or making the weight as $w = \exp\left(-\alpha\left(timestamp(B) - timestamp(A)\right)\right)$, which makes it similar to the Softmax formula. However, the softmax puts too little weight on long time intervals, so it might not be the best fit for this task. Another possible variation is that instead of using the absolute timestamp, we can only use the sequence of rating. For example, if a user has a history of A-B-C, for (A, C) we can have weight $\frac{1}{1+\alpha(2-0)}$ instead of using their timestamp. This will give us weight in the way that is more similar to the naïve model.

As a short summary of the result for this type of construction (with a small α), comparing to the naïve model result, the edges between genre gain relatively more weight, which is what we want to see. Also, the edge weights between genres get equalized a bit. Similar genres still have high weight reciprocal edges, but these weights relative to the other cross genre edge weights is less outstanding. We seem to get a more smoothed graph.

Limitations are also obvious for this model. Besides the fact that this model still does not consider rating, one biggest limitation is that it is unclear which weight function is the best for our study. Also, this model sums up the weights together to represent influence, yet one question might be "after a purchase of rock music, how long does it take for the user to likely to switch to classical music", which this model will not be able to answer. Another problem is that this model, comparing to the naïve model, runs much slower in model generation, due to the pairwise computation in each user's history.

4.3. Model 3: N-gram Model

In this model, we take a stricter approach than Model 2 in being long-sighted: we make the nodes as N-gram states to better fit the prediction job. Due to computational resource restriction, we would stop at $N = 3$ (with the 24 genres plus 1 unknown genre it creates ~15,000 nodes). The model generation procedure is similar to the above, but this time we create a bi-partite graph. Take $N = 3$, on one side we have each node representing the last three ratings of a user. On the other side we have all genres. For each user, we take their sequential rating history (tie-break arbitrarily) and accumulate their ratings as directed edges weights.

This model has a different structure from the last two models. First, it is more specialized in predicting what the user will rate next according to their history, so it is not intuitively clear how it can be used to describe users' music taste shift. This is because we are generating too many states for taste, and their meaning is unclear. However, with community detection (e.g. Louvain Algorithm), we will be able to group states together and categorize states into a smaller number of taste states. Then we accumulate the directed edges between them to get the flow between super nodes.

4.4. Model 4: Complete Model with Users

In the previous models, we have made many assumptions that might not hold true in reality and tried to build models that abstract information. It will be helpful to try building a model without these abstractions. In the previous models, we have user as an implicit factor in the graph generation process, yet now we can put them into the graph. Here, in the complete

model we will start with a bipartite graph with users and individual items (instead of categories) all as nodes, and we link directed edge from user to individual items with weight as their ratings. This will create a bipartite graph between users and items.

However, such graph cannot capture temporal information. We cannot know which rating is more likely to happen before another. As a variation, we also add links between individual items according to the users' rating history sequence (no tie-breaking: add pair wise edge for ratings at the same time). We will also need to make the edges between users and items as bidirectional for the random walk to work properly. With such variation, we add the probability that we proceed to the possible next rating, and hopefully will make the random walk cover the desired node faster. We would adjust the warp probability for the random walk according to a validation set to get the best probability of re-starting.

With similar argument as N-gram model, this model does not have a direct implication for taste state and taste change information. However, we can use community detection to recover some of them. We can run Louvain algorithm until we get a manageable size of community number, then accumulate the directed edges to analyze the flow between communities. The difference this time is that as a bonus we get communities including both items and users, by which we can classify each into different types.

5. Experiments and Model Assessment

In this section we will describe the evaluation of each models and discuss what are the problems and possible improvements for the models.

5.1. Model 1, 2, 3

These three models have similar structures and purposes, so we can evaluate them together. As assessment method, let's define a task of genre prediction problem: Given a user's rating history so far, predict the next genre they will rate.

There are 836,000 entries in the data, and each music usually has 2~3 genres. There are 24 genres in total. We sampled 5% of the entries as testing data, and we choose these testing data with preference to ratings that is in a sequence (i.e. the user has multiple ratings).

For model 1, the generation of the model did not take too long. For model 2, the generation of the model took much longer than model 1. For model 3, the generation time is between model 1 and model 2 (pairwise addition of genre sequence states was time consuming). As baseline method, we have: A. A true random predictor that picks one genre from the 24 genres. B. A predictor that always picks the most popular genre (pop in this case). We count a prediction as accurate as long as it hits one of the genres in the multiple genres. We call this accuracy "Raw Accuracy".

According to preliminary experiments, I observed that the stickiness within genres are always the strongest. This is saying that for both model 1 and model 2, they tend to predict the next genre is the same as the current genre. In order to test the performance of the model outside

this stickiness effect, I picked the test samples in which the genre we want to predict is different from the previous one and use the second highest probability to predict it. We call this accuracy "Diff Accuracy". Also, for model 3 to work to its best, we need as much history as possible. I picked the testing samples that has a history at least 3. We call this accuracy "Seq Accuracy"

The accuracies are as following in the chart:

	Baseline A	Baseline B	Model 1	Model 2	Model 3
Raw Accuracy	6.2%	9.5%	13.0%	13.0%	16.7%
Diff Accuracy	6.5%	8.7%	9.2%	9.4%	16.4%
Seq Accuracy	6.1%	10.9%	14.5%	14.5%	23.8%

As we can see from the accuracies, while the models are usually doing better than the random baseline A, baseline B is actually pretty close to what the models captures. We can see that model 1 and model 2 are almost doing the same in accuracy, which proves that the improvement in model 2 is not suffice for accuracy improvements. Model 3, the 3-gram model is obviously performing better than model 1 and 2, because it is built to specialize in such prediction job. Also, it is performing significantly better in "Seq Accuracy", because it has more information about the history.

However, since all these accuracies can be considered pretty bad in absolute performance. It means that the models are flawed or over-simplifying. There must be improvements on the models before we can rely on the model to analyze the flow of taste among users.

5.2. Model 4

Since model 4 is in a different structure, we cannot apply the same assessment procedure and compare with the previous methods. We still want to see if the model can predict the next genre. Thus, we will define a new assessment method: we perform random walk from the previous rating and see how long it will reach the target node (the next item of the rating). This time we also sample 5% of the samples as test set. We cap the largest step number as 200. We tune the probability of warping back to the original node using grid search (0, 0.1, 0.2, 0.3, 0.5) and chose 0.2. Notice that this task is significantly more difficult than the previous one.

	Bipartite Graph	Variation (with forward link)
Accuracy (200 steps cap)	16.2%	15.7%
Mean Step among Success	61.5	52.9

As result, the accuracy is not too bad for such difficult task. With the variation with adding the forward link, the accuracy shrinks a little, but the mean step to get to the target item is reduced. Considering the difficulty of the task, this model is considered a pretty good one.

6. Possible Applications

Despite of the unideal performance of the models, and while the purpose of these models and experiments are mainly to study the public's taste shift, these models do have potential

practical uses. For example, with a genre flow graph, we will be able to come up with some priors about any new music release. Existing methods of getting priors includes measuring a user's current favorite genre or measuring genre similarities, but it might gain benefit from the asymmetrical expressivity from the genre flow models, as well as the included time features. With the more advanced model, it is possible to attempt to predict a user's current favorite even when the user is offline for some amount of time. It might be helpful in winning customers back to some music app, as well as in musical advertisement personalization.

7. Conclusion and Future Work

According to the experiments, I figured out that the abstraction models (model 1, 2, 3) are too weak to produce reliable analysis about music taste shift. We might need to add additional information about rating, timestamps, user profile, etc. One aspect that gets confirmed is the stickiness within a genre. It is highly likely for any user to keeping buying music inside one genre. There are also several closely related pairs, which can be considered similar genres.

In comparison, model 4 is free from many assumptions and performs relatively well considering the difficulty of the task. It is worth exploring how to utilize such model to analyze the music taste shift of the users. It will definitely be a worth trying model no matter if we want to do prediction about future purchase or research about music taste shifts.

8. Acknowledgement

I want to send special thanks to my TA for offering constructive and helpful pointers on this project during proposal and milestone periods. Although some comments are relatively harsh, they actually provide valuable context for me to improve my methods and experiments.

9. Contribution

This project is done by Jack Mi as a single person group.

Bibliography:

Chantat Eksombatchai , Pranav Jindal , Jerry Zitao Liu , Yuchen Liu , Rahul Sharma , Charles Sugnet , Mark Ulrich , Jure Leskovec, Pixie: A System for Recommending 3+ Billion Items to 200+ Million Users in Real-Time, Proceedings of the 2018 World Wide Web Conference, April 23-27, 2018, Lyon, France [doi>10.1145/3178876.3186183]

R. He, J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. WWW, 2016

J. Leskovec, L. Adamic and B. Adamic. The Dynamics of Viral Marketing. ACM Transactions on the Web (ACM TWEB), 1(1), 2007.

H. Preda-Schimek. Modelling the Public's Taste: Local Habits, Ethnic Pluralism and European Music in Bucharest (1821–1862). Nineteenth-Century Music Review, 14(3), 391-416. doi:10.1017/S1479409817000192. 2017.