Song Recommendation Based on User's Playlist

Srivamshi Pittala Jay Patel

CS 174 Machine Learning, Course Project Report – Winter 2013

Abstract

In this course project we address the problem of providing personalized song recommendations to users. In particular, we are able to provide recommendations to users given their current playlist. Addressing this problem, we put to use some popularly used techniques of Collaborative Filtering, Content Filtering and Hybrid Filtering. We present our approach in detail and discuss the results.

Introduction

With ever increasing volume of songs becoming available on the Internet, searching for songs of interest has become a tedious task in itself. Hence a recommender system that caters to the personal interests would be of some help to the users. This task of personalized recommendations gained global attention after the advent of social media and thus machine learning has been put to good use to exploit the personal data available.

Problem Statement

Our objective for the project is to develop a recommender system that suggests songs to a user based on the songs currently in his/her playlist. We consider the user's playlist as a good indicator of the next song he/she might want to listen to. This would mean, given the contents of the playlist our recommender system should be able to provide suggestions to the playlist.

Dataset

We are using the freely available Million Song Dataset [1], which provides listening history of one million unique users. The dataset also provides a variety of metadata including audio features (like timbre, pitch, beats etc) and track information (like album, artist name, location etc) of the songs. However, due to computational limitations we chose to use only a subset of the dataset comprising of 4080 unique user's listening history and 1232 unique songs' information. We consider the following song features in this project.

Artist	Duration	Key	Loudness
Mode	Tempo	Time Signature	Year

Evaluation Criteria

For evaluating the performance of our methods, we partitioned out listening history dataset into 3500 training set and 580 testing set. Our task was now to look at the first half of the listening history available and predict the remaining half. While partitioning the dataset, we also ensured that our test set contained users with atleast two songs in their playlist. We make top 10 song recommendations to the user to validate our results. We used mAP (mean Average Precision) to evaluate our recommendations as it gives importance to the relative rankings of the recommendations rather than just checking if the recommendation is a hit or miss. Higher mAP indicates better recommendation.

Methodology

Popularity : In this method we recommend the top 10 popular songs in the training set, not in the user's playlist. We consider this as a baseline to evaluate our other methods.

Collaborative Filtering (CF) : We implemented a version each of both user-user and item-item CF methods [3]. For implementing them we built a User-Item matrix UI from the listening history information, for the training and testing set separately.

The entries of UI matrix are binary, filled as,

$$UI(i,j) = \begin{cases} 1, & if\ user\ i\ listened\ to\ song\ j \\ 0, & else \end{cases} \quad (1)$$

U/I	I_1	I_2	I_3	...	I_n
U_1					
U_2		1			
...				...	1
U_m					0

Figure 1 : Representation of UI matrix

User – User CF : In this method, users who listen to the same songs are considered to be similar and hence songs of similar users are recommended. The measure of similarity users can vary across domains. In our case, we use the cosine similarity, since it is effective when the entries of the UI are binary, at the same time being easy to compute. Higher the cosine similarity coefficient, higher is the similarity between users. The cosine coefficient between user *i* and *j* is computed as,

$$C(i,j) = \frac{U_i \cdot U_j}{\|U_i\| \cdot \|U_j\|} \quad (2)$$

The next steps are to find the top-k similar users, aggregate their playlists weighted by the cosine coefficient and recommend the top 10 songs from this aggregated playlist.

Item – Item CF : In this method, songs in the same playlist are considered to be similar and hence similar songs are recommended. We used the cosine similarity metric for the same reasons as mentioned earlier. It is computed for songs *i* and *j* as,

$$C(i,j) = \frac{I_i \cdot I_j}{\|I_i\| \cdot \|I_j\|} \quad (3)$$

Similarly, as before we prepare the aggregated playlist and recommend the top 10 songs.

Content Filtering (CN) : Our initial plan was to consider this as a binary classification task as suggested in the literature[11][12]. But we were in a difficult situation, as the data contained only what the user liked and no information of disliked songs. By the time of our project milestone we implemented a naive method of recommending songs based on overlap coefficient[5], which measures number of common features between two songs *i* and *j* as,

$$O(i,j) = \frac{n(i \cap j)}{\min(n(i), n(j))}$$

But we believed that we could employ a better approach and improve our results. There were two options available at this stage. One, to make some assumptions about a user and try to learn what might have been disliked. Second, to use only the data we have and recommend songs that are similar to the songs in the playlist. After consultation with the Professor Torresani, we chose the second option and implemented variations of kNN with metric learning.

kNN: The idea for this method is simple. Find the k nearest neighbors of every song in the visible playlist, aggregate them by their weight and recommend the top 10 songs. These 10 songs would be closest to the visible playlist.

kNN with metric learning : The problem with implementing kNN was that the dimension of our features was very large (769) and we believed that this could be reduced as most of the features were just binary vector representation of the nominal features. So we decided to implement dimensionality reduction, which is the same as unsupervised metric learning[9]. We applied linear and non-linear variants of metric learning i.e PCA and Isomap[10]. After these steps, we implemented the generic kNN as before, in reduced dimension.

Hybrid Filtering : The key motivation behind this approach is to leverage the advantages of both CF and CN. Literature[6] provided many possible ways in which this can be implemented. We chose the method called Content Boosted Collaborative Filtering (CBCF)[7], which

seemed to address the problem of data sparseness in collaborative filtering. Our user item matrix UI was mostly sparse and hence we chose to employ this method to see if we can improve our results. The main idea in this approach is to fill the vacant elements of the sparse user matrix with the recommendations from CN. This is called the pseudo user matrix (PUI), as CN approximates what the user might have liked. CF is performed on this new matrix. The entries of PUI matrix are filled as,

$$PUI(i,j) = \begin{cases} UI(i,j), & if\ UI(i,j) = 1 \\ CN(i,j), & else \end{cases} \quad (4)$$

Results and Discussion

In this section we discuss in detail the results we got by implementing the methods mentioned in previous section.

For the Popularity method we got a mAP of 0.07. This is our baseline for further comparison.

Collaborative Filtering :

Implementing the two variants for CF we got the following results for a varying k.

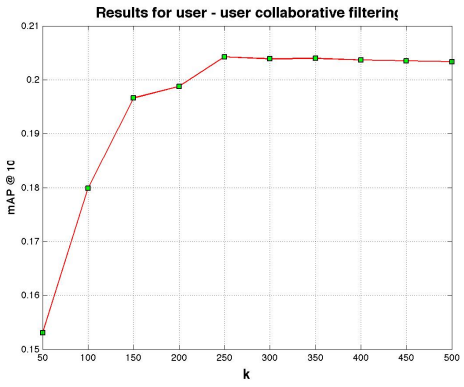

Figure : Pure user – user CF results

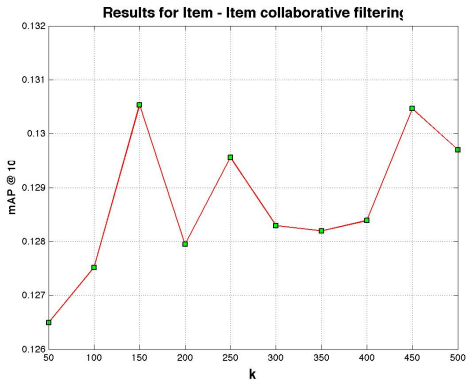

Figure : Pure item – item CF results

The user-user CF technique performed better with increase in k reaching a maximum mAP of 0.22. This increasing trend suggested a good collaboration between users. The item-item CF gave a stable mAP in the range 0.12 - 0.13 with increase in k.

Content Filtering :

Implementing CN with the overlap coefficient as a similarity criterion, we got the following poor results, where the mAP was 10 times lower than the baseline.

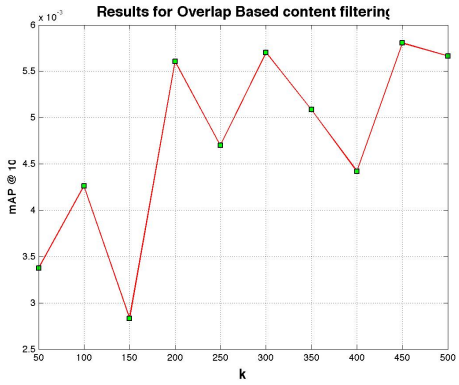

Figure : Pure CN with overlap coefficient

With a goal of improving the previous results, we implemented variants of k-NN, both with and without metric learning. The following is the result from k-NN without metric learning, where the highest mAP was 0.012 still 6 times lower than the baseline.

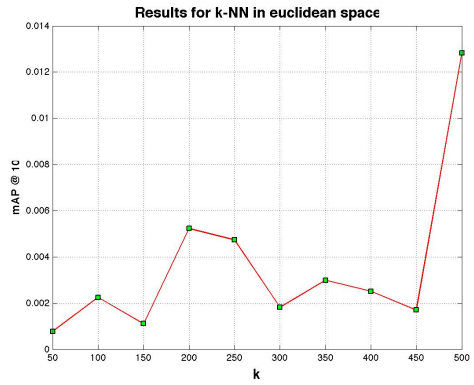

Figure : Pure CN with k-NN

Firstly, we implemented the PCA for dimensionality reduction, which is a linear method. The following is the plot for residual variance across the dimensions.

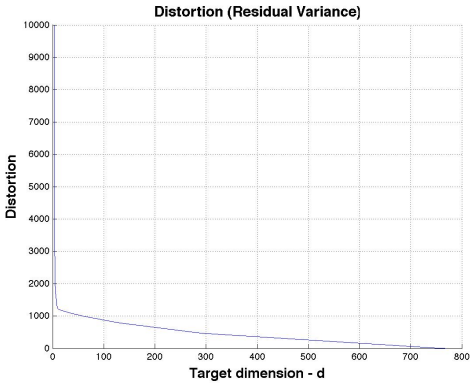

Figure : Residual variance for PCA

With PCA we were able to reduce the dimension to 4 where the captured variance was 83.03% after which there was not any significant gain in variance per increase in dimension. Applying k-NN on this reduced dimension gave the following results similar to the generic k-NN.

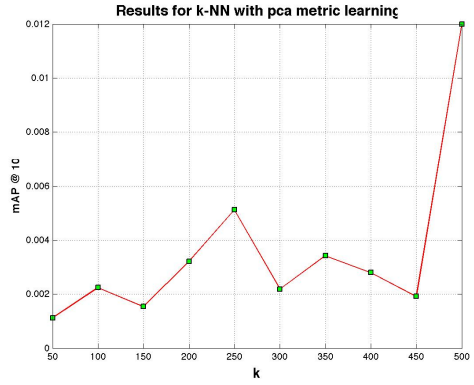

Figure : k-NN with PCA

We also implemented the Isomap, which is a non-linear dimensionality reduction method. We used the MATLAB dimensionality reduction toolbox[8] where the algorithm for Isomap is provided by them (We only used the data returned by the algorithm Isomap). The Isomap implementation gave an embedding in 2 dimensions and k-NN in this dimension also did not improve the results much.

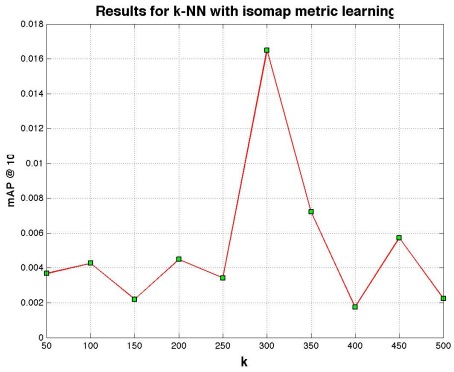

Figure : k-NN with Isomap

Hybrid Filtering *:*

We implemented the Content Boosted Collaborative Filtering method as discussed earlier. The recommendations from the Content Filtering method failed provide any boosting to the CF. Therefore the mAP remained mostly unchanged. This can be clearly attributed to the poor results from the CN methods.

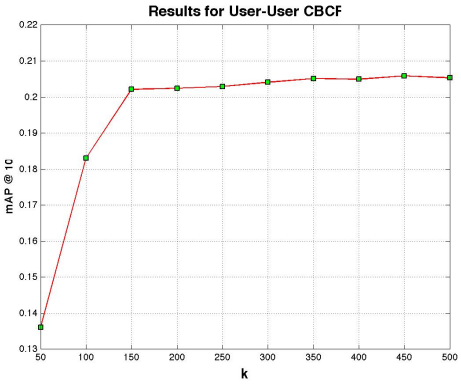

Figure : CBCF results

Conclusion

In this project, we have implemented various filtering methods addressing the problem of Song Recommendation. From our results, we can conclude that collaborative filtering outperformed all other methods. However, the low points have been the content filtering

methods, which failed to meet even the baseline. Overall, it has been a great learning experience gaining insight into what methods are being employed in the real world, which are mostly proprietary.

References

[1] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere. "The million song dataset". In Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2011).

[2] Brian McFee, Thierry Bertin-Mahieux, Daniel P.W. Ellis, and Gert R.G. Lanckriet, "The million song dataset challenge", in Proceedings of the 21st international conference companion on World Wide Web, WWW '12 Companion, pp. 909–916, New York, NY, USA, (2012). ACM

[3] Mukund Deshpande and George Karypis, "Item-based top-n recommendation algorithms", ACM Trans. Inf. Syst., 22(1), 143–177, (2004).

[4] Tianye Lu, Jing Xiong, Xiaoye Liu, "Music Recommender System Utilizing Users' Listening History and SocialNetwork Information", Machine Learning Project Report, Stanford University.

[5] Claypool M, Gokhale A, and Miranda T, "Combining Content-based and Collaborative filters in an online newspaper". In Proceedings of the SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation.

[6] Robin Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12, 4 (November 2002), 331-370

[7] Prem Melville and Raymond J. Mooney, Ramadass Nagarajan,"Content-Boosted Collaborative Filtering for Improved Recommendations". In Proceedings of the Eighteenth National Conference on Artificial Intelligence(AAAI-2002), pp. 187-192, Edmonton, Canada, July 2002

[8]http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html

[9]L. Yang, "An overview of distance metric learning". Technical report, Carnegie Mellon University, 2007.

[10] Joshua B. Tenenbaum, Vin de Silva, John C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction", SCIENCE Vol 290, 22 December 2000

[11] Meteren, R.V. & Someren, M.V. Using content-based filtering for recommendation. Structure 184, 47-56 (2000).

[12] Michael J. Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In The adaptive web, Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl (Eds.). Lecture Notes In Computer Science, Vol. 4321. Springer-Verlag, Berlin, Heidelberg 325-341.

[13] F. Aiolli. "A preliminary study on a recommender system for the million songs dataset challenge". In Proceedings of the ECAI Workshop on Preference Learning: Problems and Application in AI, Montpellier, France, 2012