Music Recommender System

Shefali Garg, 11678
Fangyan Sun EXY1329
Guide : Professor Amitabha Mukerjee

Introduction

With the rise of digital content distribution, we have access to a huge music collection. With millions of songs to choose from, we sometimes feel overwhelmed. Thus, an efficient music recommender system is necessary in the interest of both music service providers and customers.

Our study is based on Million Song Dataset Challenge in Kaggle. Our music recommender system is large-scale and personalized. We learn from users' listening history and features of songs and predict songs that a user would like to listen to.

Dataset

We are mainly using 2 datasets.

1. Data A: Dataset provided by Kaggle: users ID, songs ID and triplets (user,song,count)
   - 1,200,000 users, more than 380 000 songs and 48 million triplets gathered from users' listening histories in total
   - We only work on 10,000 users' listening history.
   - We create a Matrix M from the triplets.
2. Data B: Feature files extracted by ourselves from meta data of song from the website of *labrosa.ee.columbia.edu/millionsong/*
   - 280 GB of meta data
   - Each song is represented by a feature vector of 10 components including year, duration, loudness, artist, danceability, etc.
   - Due to memory limitations, we ony get features of 10,000 songs(3 GB)

Popularity based Model

Idea

1.Sort songs by popularity in a decreasing order

2.For each user, recommend the songs in order of popularity, except those already in the user's profile

- ❖ Simple, easy, popular songs are listened widely.
- ❖ Not personalized
- ❖ Some songs will never be listend

mAP = 2.0138 %

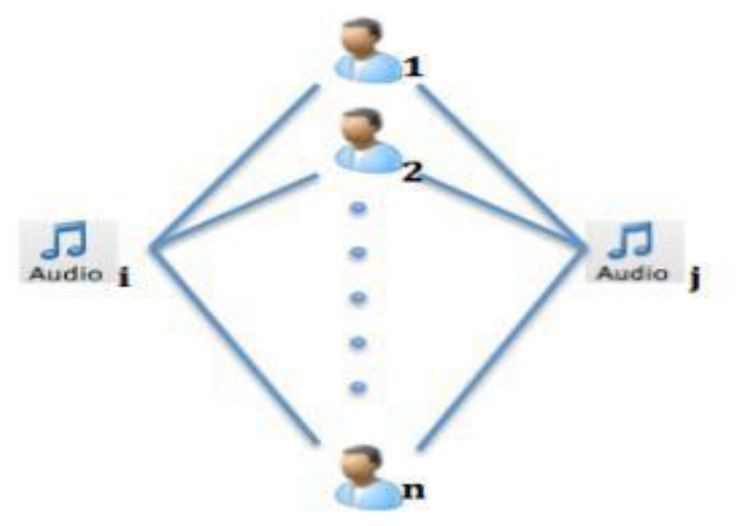
Collaborative based Model

Idea 1

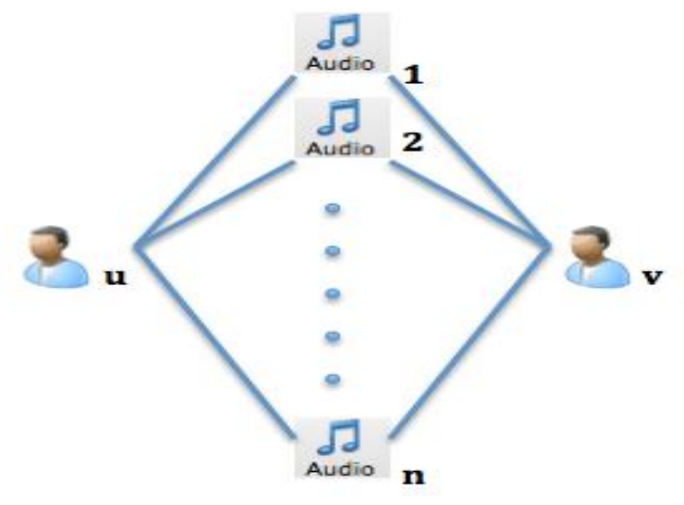
•Songs that are often listened by the same user tend to be similar and are more likely to be listened together in future by some other user.

Idea 2

•Users who listen to the same songs in the past tend to have similar interests and will probably listen to the same songs in future.



Song based Model



User based Model

e.g. user-based

Conditional probability measure of similarity between two users:

$$W_{u,v} = \mathrm{P}(\mathrm{v}|\mathrm{u})^{\alpha}.\mathrm{P}(\mathrm{u}|\mathrm{v})^{1-\alpha} \text{ with } \alpha \epsilon\ [0,1]$$

Locality of scoring function:

- Emphasize similar items, determine how individual scoring component influences overall scoring: $f(w) = w^q \; with \; q \; \in N$

Stochastic aggregation of two lists, randomly chooses one of lists according to probability distribution over predictors and recommends best scored items of lists not yet inserted in final recommendation.

Remark

- When the song history of a user is too small to leverage the power of the user-based recommendation algorithm, we can offer recommendations based on song similarity, which yield better results with smaller song histories
- Using play count does not give good result because similarity model biased to few songs played multiple times, calculation noise is generated by a few very popular songs.
- There's no personalization and majority of songs have too few listeners

IS with (α = 0.15, *q* = 3)
US with (α = 0.3, *q* = 5)
mAP(Stochastic) = = 8.2117 %

SVD Model

Idea

•Listening histories are influenced by a set of factors specific to the domain (e.g. Genre, artist...)

•Users and songs characterized by latent factors.

Decomposes M into a latent feature space that relates users to songs

$$M = U\,.\Sigma.V$$

with M ∈ R^{m*n} ,$U \in R^{k*m}$,$\Sigma \in R^{k*k}$and V ∈ R^{k*n}

•U is the user factor while V represents song factors

•For each user, a personalized recommendation is given by ranking the following item for each song:

$$W_i = U_u^T\,.V_i$$

mAP = 3.18 %

Analysis

•There is not enough data for the algorithm to arrive at a good prediction. The median number of songs in a user's play count history is fourteen to fifteen, this sparseness does not allow the SVD objective function to converge to a global optimum

KNN Model

Idea

•From data B, we create a feature space of songs (features are normalized)

•In this space, we find the k nearest neighbors for each song by calculating their Euclidean Distance

•Look at each user's profile and suggest songs which are their neighbors

mAP = 0.6867 % for k = 50

Evaluation Metric

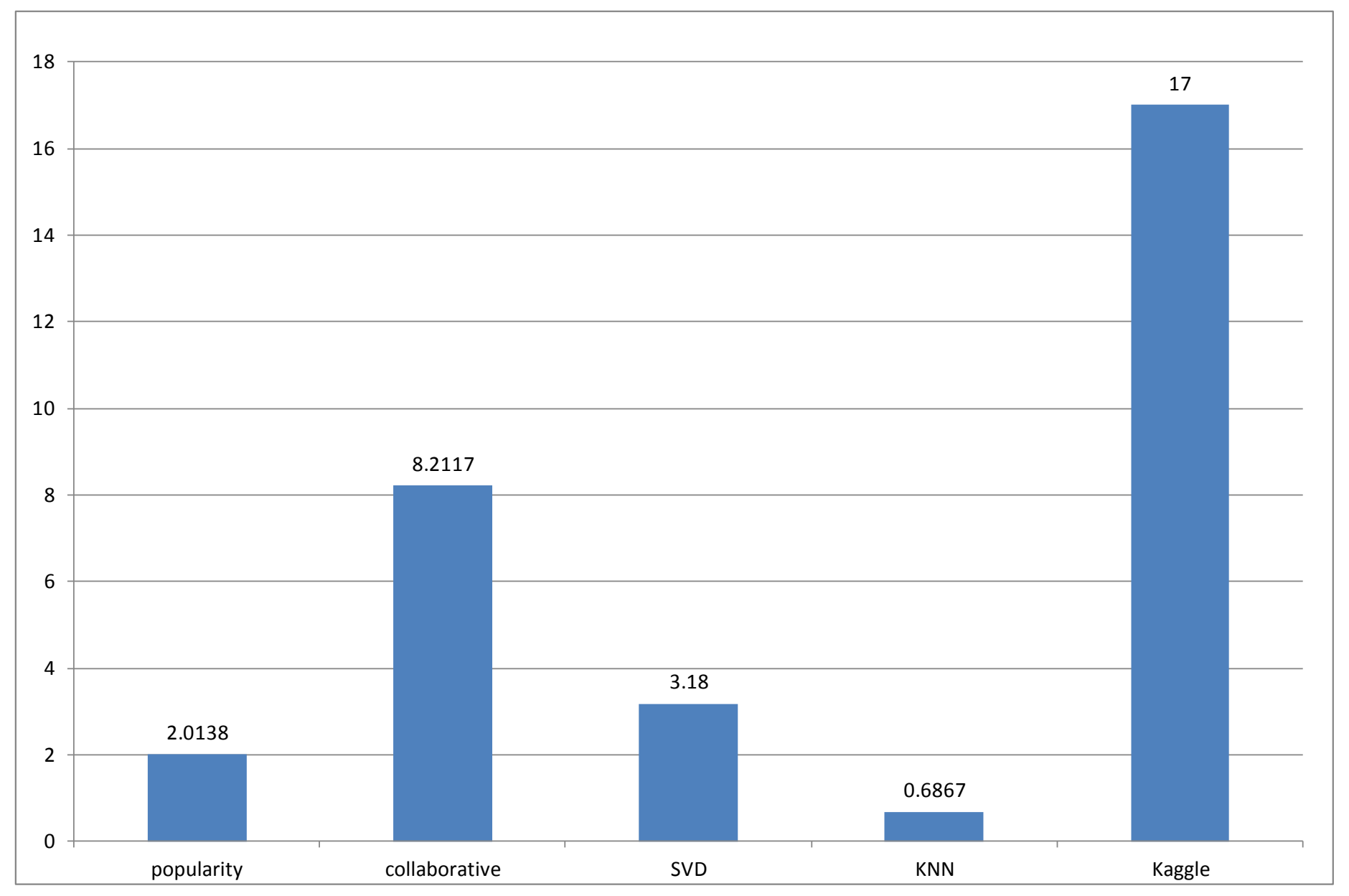
Mean average precision(mAP)

•Proportion of correct recommendations with more weight to top ones

•precision is much more important than recall because false positives can lead to a poor user experience

1. Precision at k: proportion of correct recommendations within the top – k of the predicted ranking: $$P_k(u,y) = \frac{1}{k}\sum_{j=1}^{k} M_{u,y}(j)\ ,\ \forall k \le t$$
2. For each user, the average precision at each recall point: $$AP\,(u,y) = \frac{1}{n_u}\sum_{j=1}^{t} P_k(u,y)\,.\,M_{u,y}(k)$$
3. Mean average precision: $\mathrm{mAP} = \frac{1}{m}\sum AP\left(u, y_u\right)$

Results



Winner Fabio Aiolli got mAP of 0.1791 for the top 500 songs

Conclusion

Building a recommender system is not a trival task. The fact that it's large scale dateset makes it difficult in many aspects.

1.Recommending 500 « right» songs out of 380 million songs for different users is not easy to get a high precision. That's why we didn't get any result better than 10 %. Even the Kaggle winner has only got 17 %.

2.The meta data includes huge information and when exploring it, it is difficult to extract relevant features for song.

3.Processing such a huge dataset is memory and CPU intensive.

Future work

- ❖ Run the algorithms on a distributed system, like Hadoop or Condor, to parallelize the computation, decrease the runtime and leverage distributed memory to run the complete MSD.
- ❖ Combine different methods and learn the weightage for each method according to the dataset
- ❖ Automatically generate relevant features
- ❖ Develop more recommendation algorithms based on different data (e.g. the how the user is feeling, social recommendation, etc)

REFERENCE

[1] MCFEE, B., BERTINMAHIEUX,T., ELLIS, D. P., LANCKRIET, G. R. (2012, APRIL). *THE MILLION SONG DATASET CHALLENGE*. IN PROCEEDINGS OF THE 21ST INTERNATIONAL CONFERENCE COMPANION ON WORLD WIDE WEB (PP. 909916).ACM.
[2] AIOLLI, F. (2012). *A PRELIMINARY STUDY ON A RECOMMENDER SYSTEM FOR THE MILLION SONGS DATASET CHALLENGE*. PREFERENCE LEARNING: PROBLEMS AND APPLICATIONS IN AI
[3] KOREN, YEHUDA. *"RECOMMENDER SYSTEM UTILIZING COLLABORATIVE FILTERING COMBINING EXPLICIT AND IMPLICIT FEEDBACK WITH BOTH NEIGHBORHOOD AND LATENT FACTOR MODELS."*
[4] CREMONESI, PAOLO, YEHUDA KOREN, AND ROBERTO TURRIN. "PERFORMANCE OF RECOMMENDER ALGORITHMS ON TOP-N RECOMMENDATION TASKS." *PROCEEDINGS OF THE FOURTH ACM CONFERENCE ON RECOMMENDER SYSTEMS*. ACM, 2010