Music Recommendation System: Final Presentation

- Team:
- Vahe Avagyan
- Kartik Saxena
- Prateek Verma

Problem Statement

- Implementation of a music recommendation system.

Given a User U , Playlist = { S1 , S2 … Si } Predict ->S(i+1).

<u>Challenges</u>

- No straightforward metric to predict similarity between two songs. Two similar songs that sound similar(Audio Features) may have completely different audiences.

- Lack of direct feedback from a user about an item (rating) . We just have information whether a user listened to a song or not, not about how much they liked it.

Dataset

- **Million Song Dataset:**
  - SecondHandSongs dataset -> cover songs
  - musiXmatch dataset -> lyrics
  - Last.fm dataset -> song-level tags and similarity
  - Taste Profile subset -> Echonest Taste-Profile Data:
    - Song History of 1 Million users for ~ unique 300,000 Songs
    - Consist of 48 million <User – Song – PlayCount> triplets

userID	SongID	Count
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOAKIMP12A8C130995	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOAPDEY12A81C210A9	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SODDNQT12A6D4F5F7E	5
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBSUJE12A6D4F8CF5	2
….	…	…

Data Representation

userID	SongID	Count
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOAKIMP12A8C130995	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOAPDEY12A81C210A9	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SODDNQT12A6D4F5F7E	5
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBSUJE12A6D4F8CF5	2

Represent it as a matrix

Unique SongIDs

Unique UserIDs

1	1	5	2
1	0	6	0
0	1	0	3

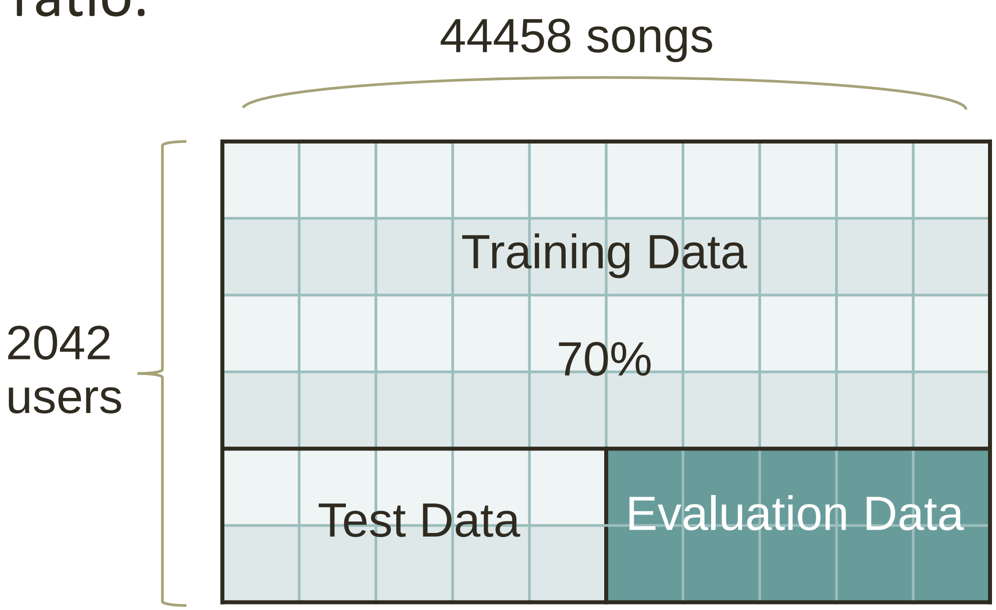
Play Counts of Song *j* for user *i*.

Convert to sparse and normalize

```
(4,32)    0.0654
(1,34)    0.0217
(4,51)    0.0093
(3,57)    0.0118
(2,69)    0.0063
(3,72)    0.0118
(4,81)    0.0093
(2,108)   0.0252
   ...
```

Data Set: Training & Testing

- 100k tuples of the EchoNest Data was used,
- contains listening history of 2042 Unique users and 44458 Unique songs.
- 2042 users were split into training and testing sets in 70:30 ratio.



Models

- User based collaborative filtering using K-Nearest Neighbor KNN.
  - Choosing user with largest play history
  - Aggregating songs from all the K users
- User based using cosine similarity.
- Item based collaborative filtering using cosine similarity.
- User based collaborative filtering using hierarchical clustering.
- Mixed Model – Combination of user based and item based collaborative filtering.

User-Based Collaborative Filtering using K -Nearest Neighbor (KNN)

- We use KNN algorithm to select k similar users to the sample user based on cosine distance.

$$sim(x,y) = cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{||\vec{x}||_2 \times ||\vec{y}||_2} = \frac{\sum\limits_{i\in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum\limits_{i\in I_{xy}} r_{x,i}^2}\sqrt{\sum\limits_{i\in I_{xy}} r_{y,i}^2}}$$

Song histories are feature vectors, where play counts are considered as weights for a particular song.

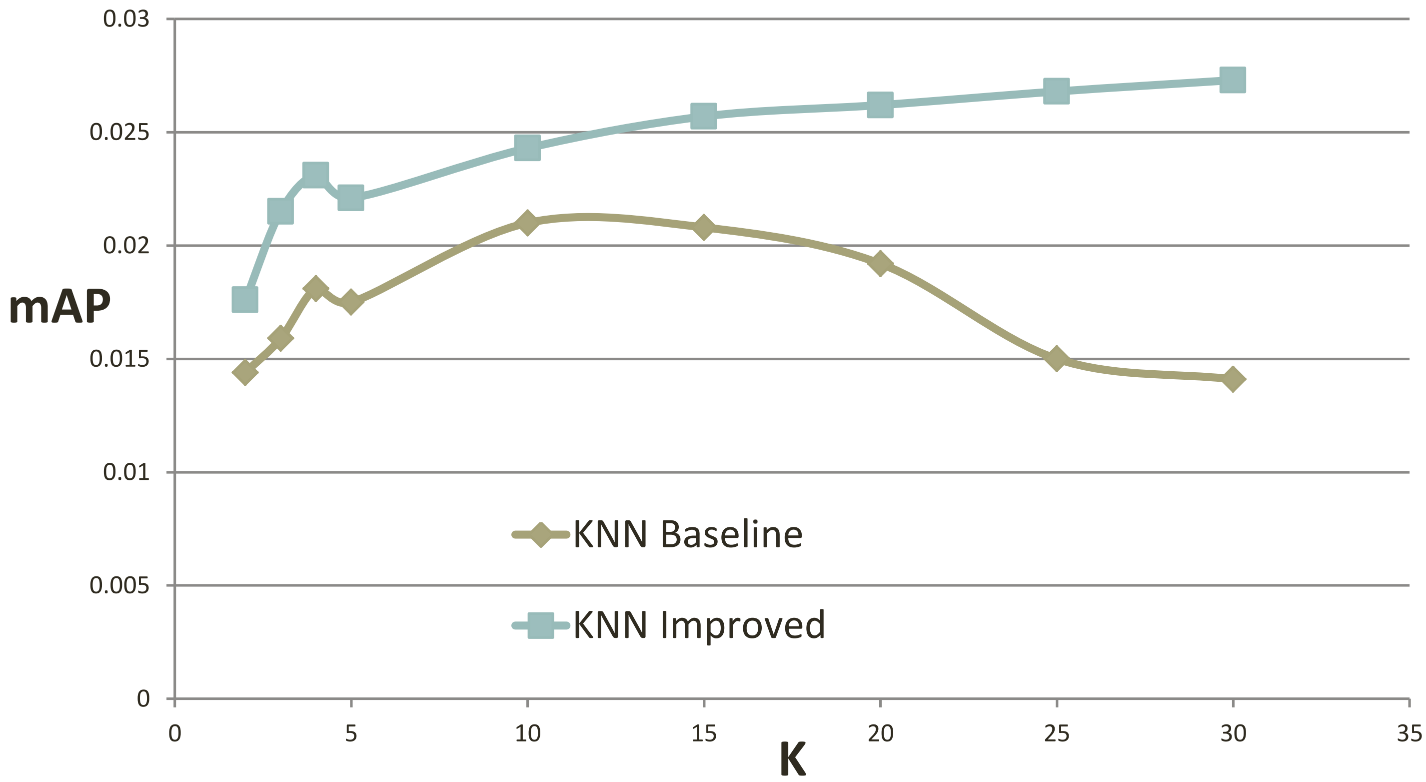
Selection of songs for recommendation

- Baseline: We selected among the K users, the user having largest play history.
  - Not a good heuristic.
  - Unpredictable result with increasing K.

- Improved: From the play list of K similar users, we aggregate all the songs, and sort it using play count and recommend only top N songs.
  - Improves with increasing value of K.

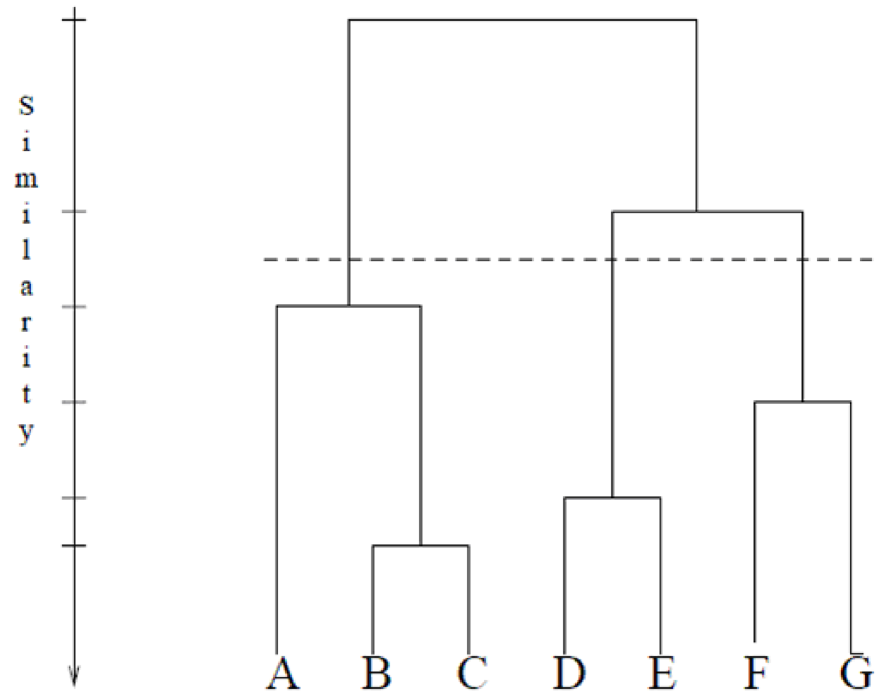
Improved Recommendation using KNN

- Comparison of baseline and improved KNN algorithms



User based collaborative filtering using hierarchical clustering

- We use agglomerative clustering
  - "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.



User based collaborative filtering using cosine similarity

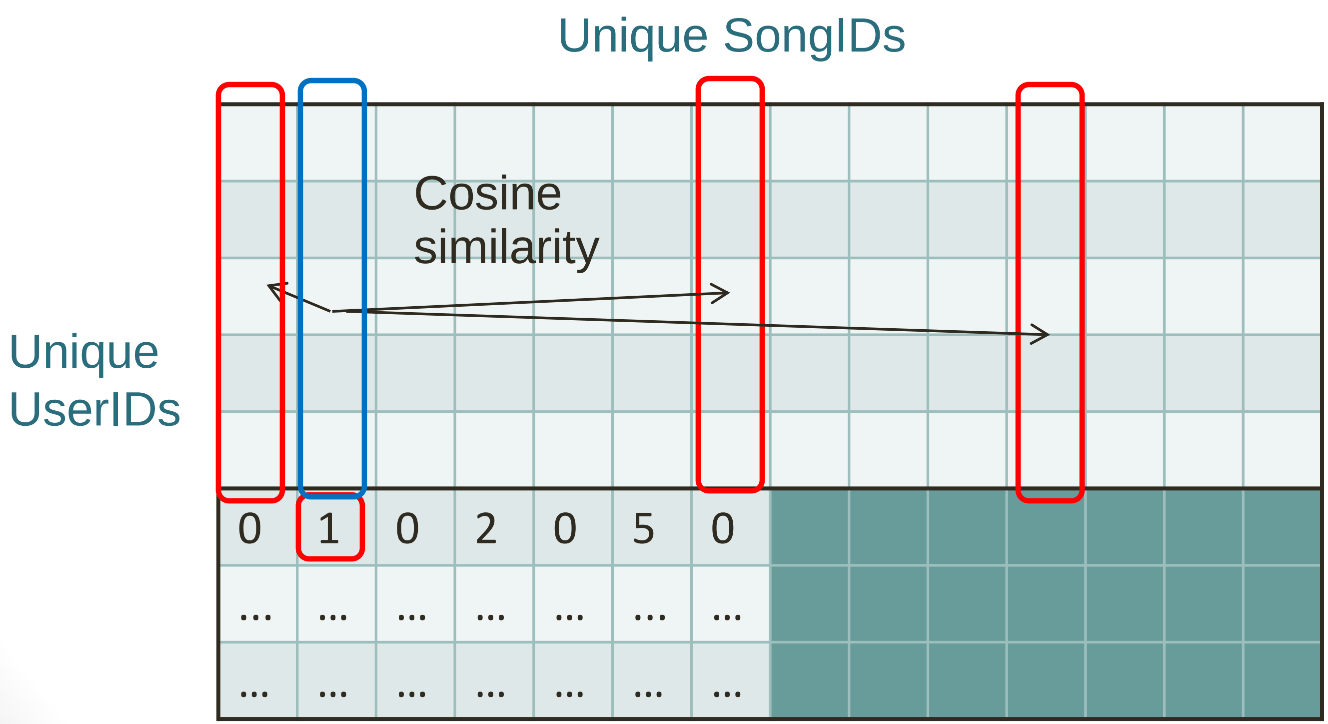
- Compute similarity between users using cosine similarity: Where vector of listening history is feature for each users.

- It is different from KNN as K similar users are assigned different scores based on similarity unlike KNN.

$$sim(u,v) = \frac{\#\text{common items}(u,v)}{\#\text{items}(u)^{\frac{1}{2}} \cdot \#\text{items}(v)^{\frac{1}{2}}}$$

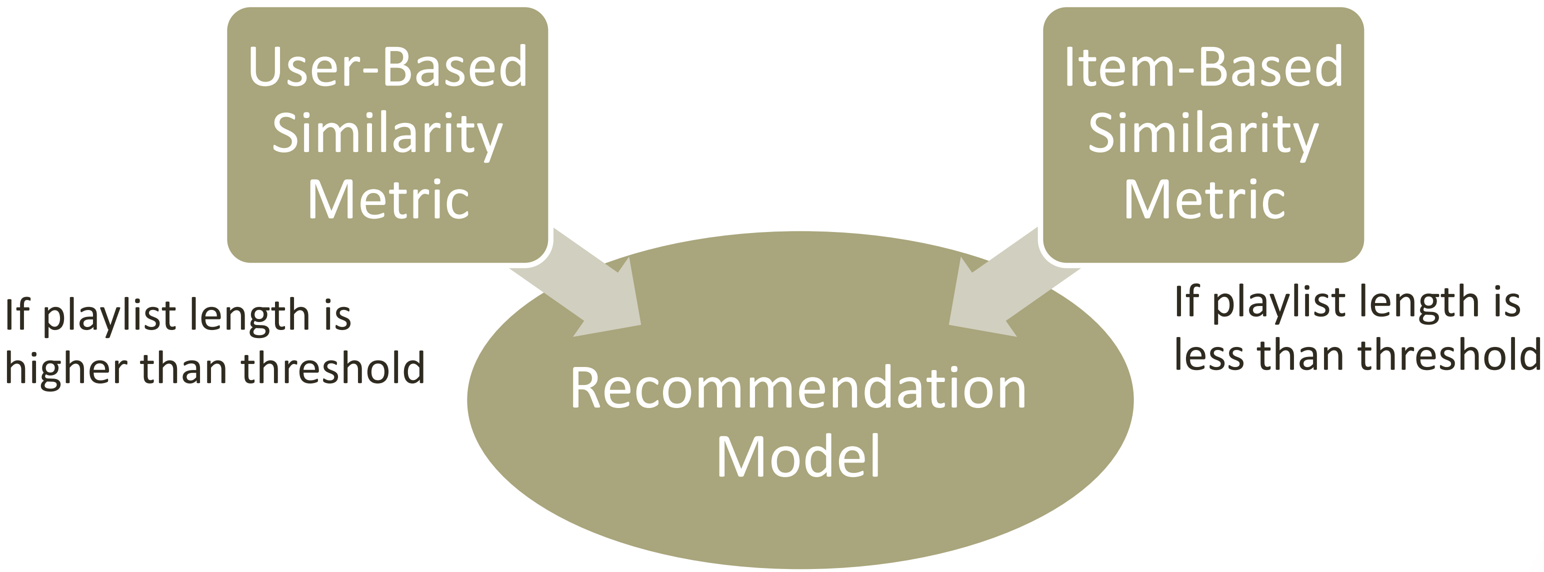
Item based collaborative filtering using cosine similarity

- Compute the most similar items for the items that have been already listened by the active user, and then aggregate those items to form the final recommendation.



Mixed Model

- We combine Item based and User based collaborative filtering based on playlist length



Evaluation

mAP (Mean Average Precision) as an evaluation metric.

$$\pi_k(u, y) = \frac{1}{k} \sum_{p=1}^{k} r_{uy(p)}$$

y denotes ranking over the recommended songs.

$y(p)=i$ means that the item i is ranked at position p.

The precision for k, $Pi(k)$ is defined as the proportion of correct recommendations within the top-k of the predicted rankings.

$$AP(u, y) = \frac{1}{\tau_u} \sum_{p=1}^{\tau} \pi_k(u, y) r_{uy(p)}$$

Average precision is the average of the precisions at each recall point, where τ_u is the smaller between τ and the number of user u's positively associated songs.

Mean average precision is the average of $AP(u,y)$ over all the users.

MAP Evaluations

	KNN Baseline	KNN Improved	Hierarchical Clustering	User based cosine	Item based cosine	Mixed model
mAP	0.0141	0.0273	0.0167	0.0348	0.0015	0.0166

Scalability and Complexity

- KNN , Collaborative Filtering Algorithms can be implemented on top of Apache Mahout library which is based on the Map-Reduce paradigm. http://mahout.apache.org/
- Map-Reduce can be also used for pre-processing and dataset creation.
- Challenging to work with such a big data-set , hence we chose to work on a subset of it[First 100k out of 48Million Entries].

	KNN Baseline/ Improved	Hierarchical Clustering	User based cosine	Item based cosine	Mixed model
Complexity	$O(N^2)$	$O(N^3)$	$O(N^2)$	$O(M^2)$	$O(M^2)$

N – Number of users in the table,
M – Number of songs,
N << M

Conclusions/Lessons learnt

- Item based similarity:
  - Limited number of users => No full representation of song features across many users.
  - As number of songs are much larger than users, hence running cosine similarity takes much longer.
- Mixed model:
  - Results for user based collaborative filtering does not depend on significantly on the length of play list and therefore mixture model does not improve the result.
  - Results for the mixture model were also influenced by the lower results obtained by the item based collaborative filtering.
- Hierarchical Clustering:
  - Gives similarity of users in general, therefore mAP value is less.

- Thanks

Questions?