Music Recommendation System

Million Song Dataset from Kaggle

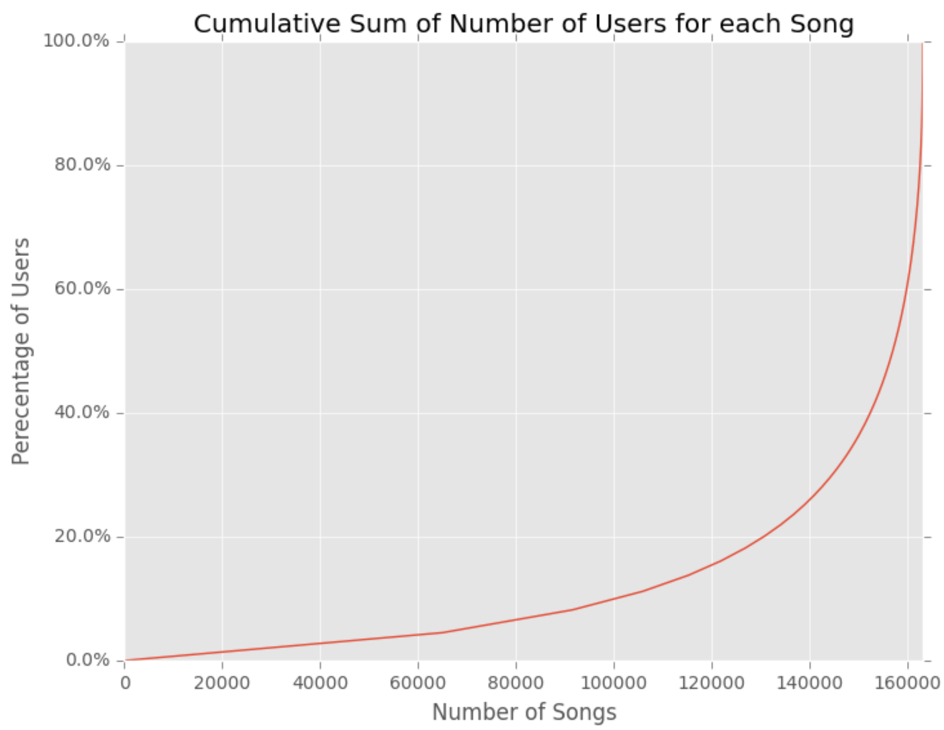
Erin Grand, Justin Law, Jordan Rosenblum

Introduction

- Recommendation systems allow for users to discover things of interest, ideally with zero to little effort
- We focused on collaborative filtering approaches without the need for metadata
  - User feedback (i.e. number of plays) is used instead
- We explored matrix factorization and user and item based collaborative filtering

Data and Statistics

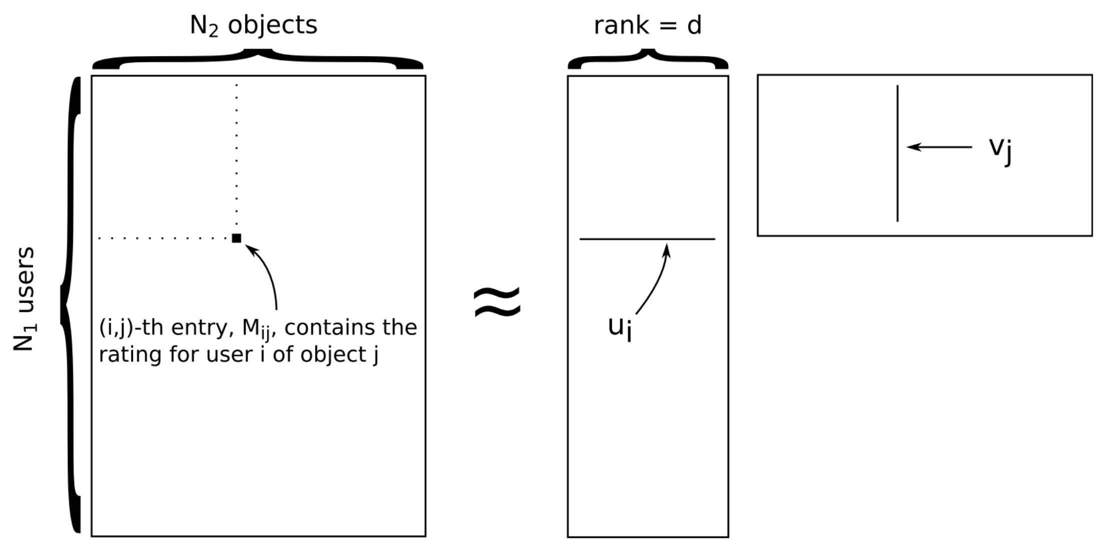
- The sample data set had 110,000 users and 163,206 songs
- We subset the data to users with > 27 songs played and songs with > 22 users listening to them -> 8,130 users and 11,861 songs (10% of orig.)
  - Sparsity was 0.16%



Evaluation and Baseline

- Benchmarked our algorithms using a mean average precision score truncated at 500 recommendations
- MAP@500 takes into account the first 500 recommendations given to each user and calculates the number of true positives vs. the test set against the total number of recommended songs at each position, averaged together. It then takes the average across all users
- We used two baseline methods, recommending:
  - the 500 most popular songs (MAP = 0.0138)
  - the most popular songs from the artists that the user had already listened to (MAP = 0.0448)

Matrix Factorization
Hyperparameters



Alternating least squares optimization

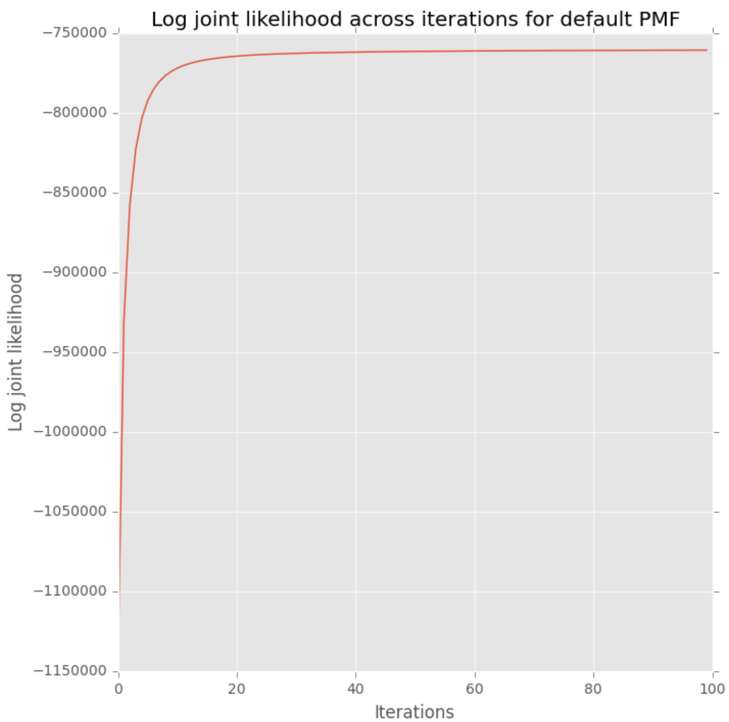
$$u_i = \left(\lambda\sigma^2 I + \sum_{j \in \Omega_{u_i}} v_j v_j^T\right)^{-1} \left(\sum_{j \in \Omega_{u_i}} M_{ij} v_j\right)$$

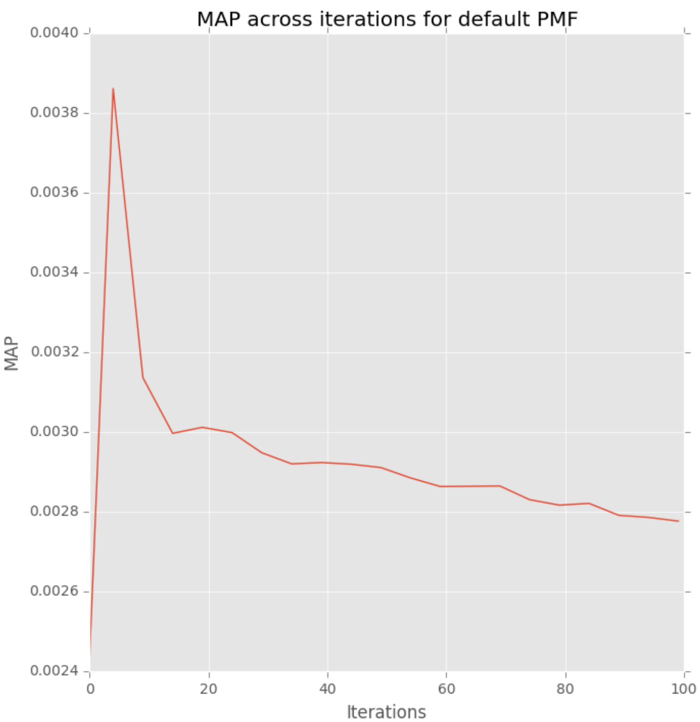
$$v_j = \left(\lambda\sigma^2 I + \sum_{i \in \Omega_{v_j}} u_i u_i^T\right)^{-1} \left(\sum_{i \in \Omega_{v_j}} M_{ij} u_i\right)$$

Hyperparameters

- Rank
- Variance
- Iterations
- Lambda set to 10

Matrix Factorization
Initial Results



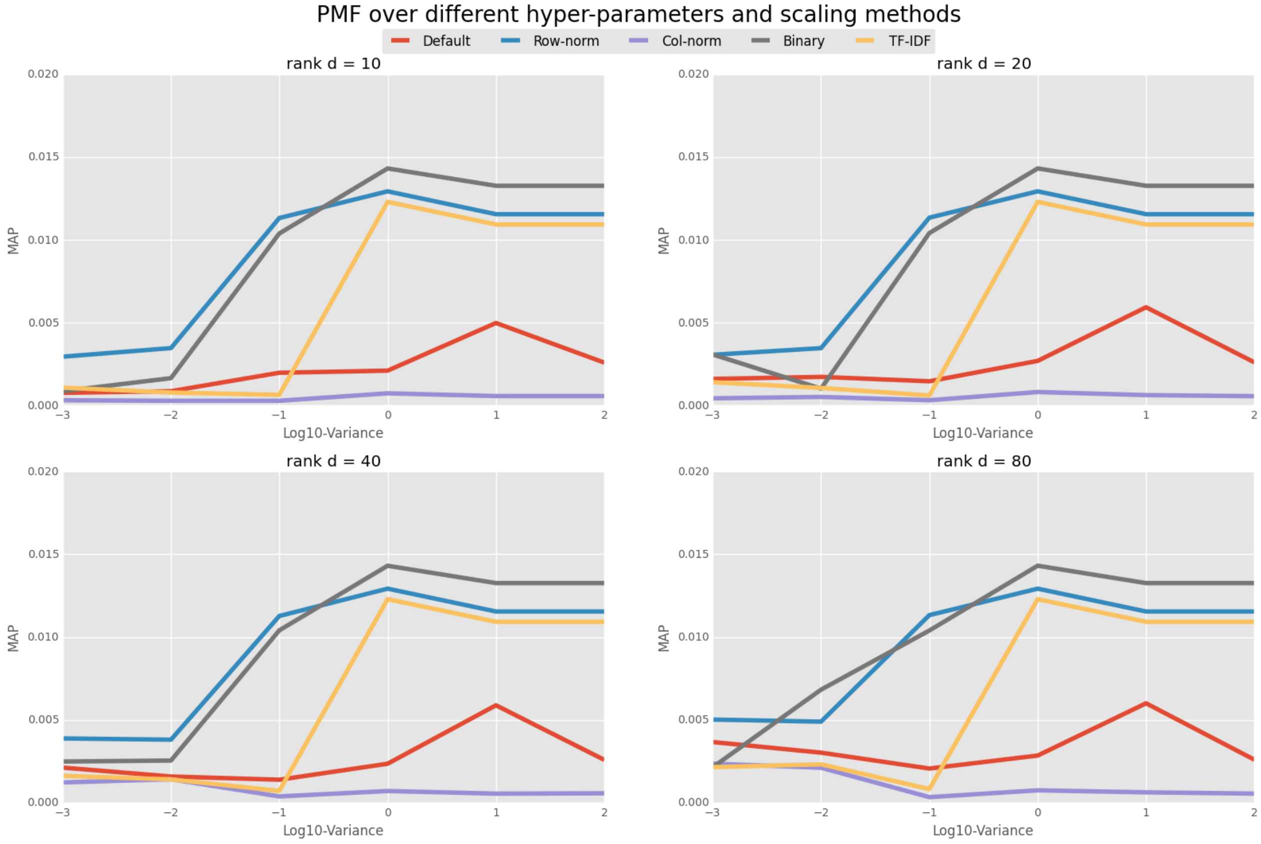


Using rank = 80 and variance = 1.0 with play count values

- MAP values were significantly below the popularity baseline
- MAP values were poor and do not change significantly across iterations
- Log joint likelihood converged sufficiently within 30 iterations
- Implicit feedback issue required testing of normalization schemas

Matrix Factorization

Normalization Schemas



- Rank had marginal impact
- Variance had some correlation with the normalization used
- Binary scheme provided the best MAP value of 0.0143
- Data was likely too sparse for matrix factorization to identify biases

User-based CF

- Calculated similarity of every pair of users in the subset

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

- Determined the weight on each song, *i*, for a particular user, *u*, by summing the similarity scores between user *u* and all users *v* who listened to song *i*

$$w_i = \sum_{v \in V} sim(u,v)$$

- Recommend songs with highest weights in descending order

Item-based CF

- Calculated similarity of every pair of songs in the subset

$$sim(i,j) = \frac{\#\ \text{common users}(i,j)}{\#\ \text{items}(i)^{1/2} \times \#\ \text{items}(j)^{1/2}}$$

- For each song, *b*, that was found to be similar to one of the songs, *a*, that a user listened to, we calculated the weight for that song: since the same song will likely come up multiple times for a different song, *a*, we then summed similarity scores for each similar song, *b*, across all songs, *a*

$$w_b = \sum_{a \in A} sim(a,b)$$

- Recommend songs with highest weights in descending order

Results and Next Steps

Algorithm	MAP score
Item Based CF	0.0479
Artist Based Popularity Baseline	0.0448
User Based CF	0.0377
Binary Matrix Factorization	0.0143
Top 500 Songs by Count Baseline	0.0138
Top 500 Songs by Plays Baseline	0.0126

- Next Steps
  - Incorporate tags and metadata present in the dataset (e.g. year, genre, audio metadata)
  - Expand the size of the subset and distribute the workload to multiple machines

Matrix Factorization: Equations

$$u_i = \left(\lambda\sigma^2 I + \sum_{j\in\Omega_{u_i}} v_j v_j^T\right)^{-1} \left(\sum_{j\in\Omega_{u_i}} M_{ij} v_j\right)$$

$$v_j = \left(\lambda\sigma^2 I + \sum_{i\in\Omega_{v_j}} u_i u_i^T\right)^{-1} \left(\sum_{i\in\Omega_{v_j}} M_{ij} u_i\right)$$

$$\mathcal{L} = -\sum_{(i,j)\in\Omega} \frac{1}{2\sigma^2}||M_{ij} - u_i^T v_j||^2 - \sum_{i=1}^{N_1} \frac{\lambda}{2}||u_i^2|| - \sum_{j=1}^{N_2} \frac{\lambda}{2}||v_j^2|| + \text{constant}$$