Million Song Dataset

...

Will Vining
Chris Ottino
Taylor Berger
Aaron Gonzales

Collaborative Filtering

- Used Mahout
- Pre-Normalization : RMSE = 178 (oops)
- Normalization is done on a per user basis where we take current song plays / total song plays for that user
- Output the song with highest predicted “normalized” play count

CF: RMSE

Similarity Method	Training Percentage = 0.7
Pearson	0.21018
Spearman	0.210703
Euclidean Distance *	0.101453

Content-Based Headaches

- Used Last.fm song similarities / genre tags
  - tags = top 500 genre tags

- built similar song set for each user’s top 2 songs *from last.fm similar song data*

cb preprocessing

combined categorical features with audio features:

a. genre tags, loudness, tempo, key, mode, “hotttnesss”, familiarity, time, duration
    i. 500~ dimensional dummy-coded vectors
    ii. audio features from original MSD using AWS
b. calculated similarity tables for each song in a user’s similar song set made from last.fm similars data (~~100 songs per user)
c. similarity(s1, s2) was a combination of Jaccard and cosine similarities for categorical and numeric features

(this took a while)

final recommendations

```
calculate cb_recommendations


for user in users:
    cf_pred = cf_pred_val(user)
    cb_pred = cb_pred_val(user)


    if cb_pred > cf_pred && cf_pred > 0.1:
        return cb_rec
    else return cf_rec
```

"RMSE"

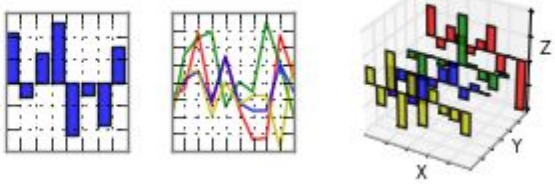
- completely arbitrary
- predicted value was *"normalized" play count*

System	RMSE
CF Only	0.101453
CB+CF	0.13765

Tools

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



MILLION SONG DATASET

THE UNIVERSITY of NEW MEXICO | Center for Advanced Research Computing

Spark

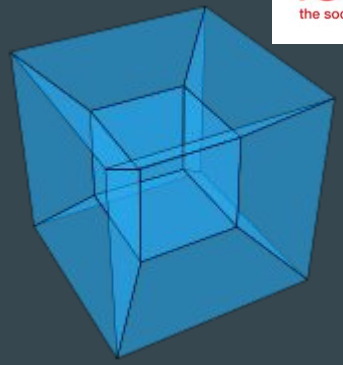
jupyter

NumPy

the echonest

last·fm
the social music revolution

UNM | Computer Science



mahout

powered by amazon web services™

Java

Google Cloud Platform

python™