Music Recommendation Systems

Emilien Dupont, Isabelle Rao, William Zhang
ICME, MS&E, Stanford University

Recommending Music

Companies like Spotify and Pandora recommend new songs to their users based on the actual content or attributes of the songs and user listening histories. The goal of this project is to recommend songs based solely on **user histories** with no meta-information about the songs.

Data

The data comes from the *Million Song Dataset*. It is in the form of (user, song, song count) triplets, for example:

(Isabelle, Hey Jude, 5)
(Isabelle, Shake It Off, 13)
(William, Whole Lotta Love, 10)
(Emilien, Shake It Off, 32)

- 48 million triplets
- 380K songs
- 1 million users

- Avg # of listens per user: 142
- Avg # unique songs per user: 50
- Highest song count: 9667

Store data in a *user x item* matrix of counts *C:*

$$C = \begin{pmatrix} 5 & 0 & 13 \\ 0 & 10 & 0 \\ ? & ? & 32 \end{pmatrix}$$

Count to Rating

Many algorithms for recommender systems work best with explicit ratings. How to convert song count to something that mimics rating?

1. Normalize by the max: $R(u,i) = \frac{C(u,i)}{\max_i C(u,i)}$
2. Average: $R(u,i) = \frac{C(u,i)}{\sum_i C(u,i)}$
3. Binary: $R(u,i) = \begin{cases} 1 & if\ C(u,i) > 0 \\ 0 & otherwise \end{cases}$
4. Exponential of max normalized: $R(u,i) = \exp\left(\frac{C(u,i)}{\max_i C(u,i)}\right)$

Collaborative filtering methods

- Baseline predictor: recommend 500 most popular songs.
- Neighborhood model (cosine similarity)
- Latent factor model

Neighborhood Model

Let u, v denote two users; i, j denote two items; $\mathrm{U(i)}$ is the set of users who have listened to song i; and $\mathrm{I(u)}$ is the set of songs user u has listened to.

- In user/user CF: $similarity(u,v) = \frac{r_u \cdot r_v}{\|r_u\|_2 \|r_v\|_2}$.
- In item/item CF: $similarity(i,j) = \frac{r_i \cdot r_j}{\|r_i\|_2 \|r_j\|_2}$.

Once we have defined the similarity matrix, we can make recommendations using **score functions**:

- User-based function: $score(u,i) = \sum_{v \in U(i)} f\big(similarity(u,v)\big)$
- Item-based function: $score(u,i) = \sum_{j \in I(u)} f\big(similarity(i,j)\big)$

where $f(x)$ is the scoring function.

Latent Factor Model

Goal: to uncover *k* "latent factors" that explain observed ratings.
One particular flavor inspired by the **SVD**: $R \approx X^T Y$ where X^T, Y are "tall" and "fat" matrices.
Predicted score: $\hat{r}_{u,i} = x_u^T y_i$ - choose 500 largest scores
Optimization problem:

$$\min_{X,Y} \sum_{\text{known } r_{u,i}} \left(r_{u,i} - x_u^T y_i\right)^2 + \lambda_x \|x_u\|^2 + \lambda_y \|y_i\|^2$$

Algorithm: Alternating Least Squares
Repeat until convergence {

for each user u: $x_u \leftarrow \left(\sum_{r_{u,i} \in r_{u*}} y_i\, y_i^T + \lambda_x I_k\right)^{-1} \sum_{r_{u,i} \in r_{u*}} r_{u,i} y_i$

for each user i: $y_i \leftarrow \left(\sum_{r_{u,i} \in r_{*i}} x_u\, x_u^T + \lambda_y I_k\right)^{-1} \sum_{r_{u,i} \in r_{*i}} r_{u,i} x_u$

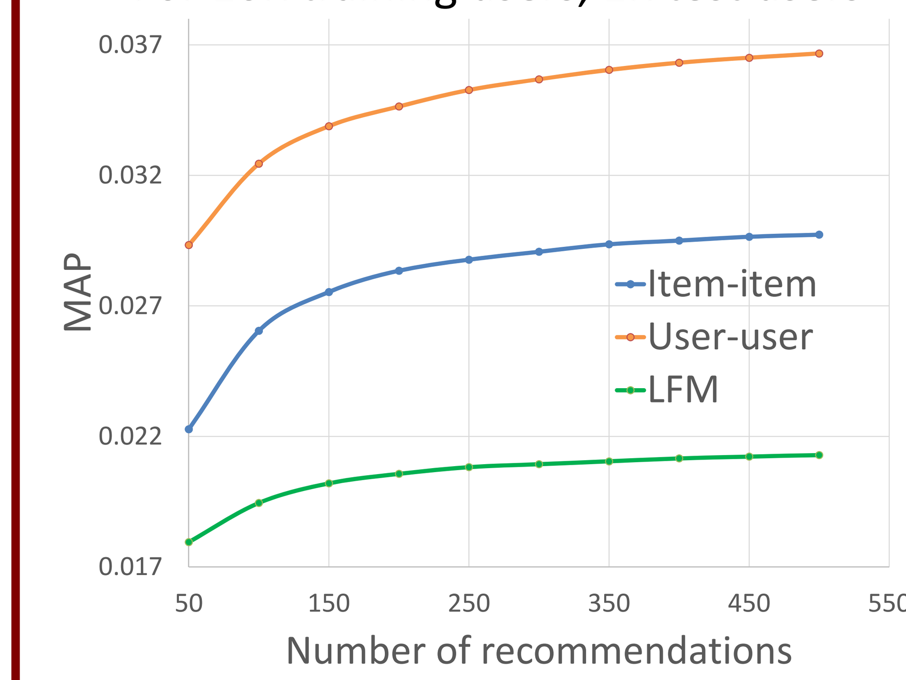
}

Results and visualization

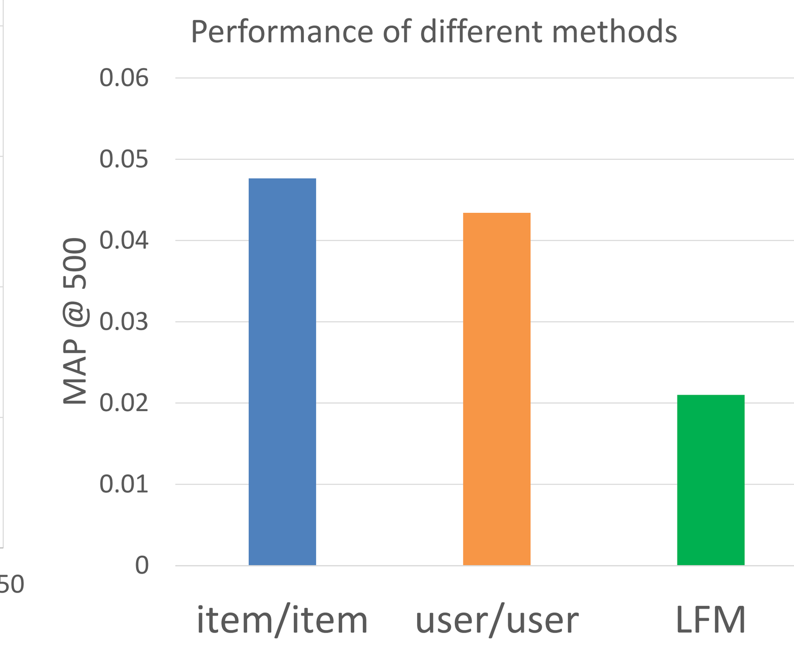
Metric to evaluate results: **Mean Average Precision** (MAP). Given a vector y of recommendations, sorted from most to least confident, and user u:

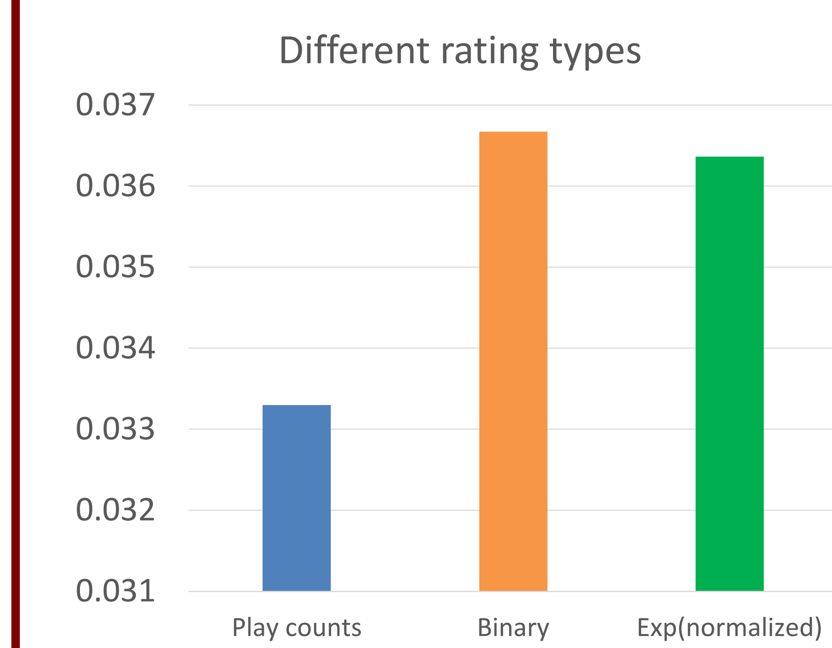
- Precision at k: $P_k(u,y) = \frac{1}{k}\sum_{j=1}^{k} \mathbb{I}\{y(j) \text{ is in } u\text{'s hidden history } \mathcal{H}(u)\}$
- Average precision: $\mathrm{AP}(u,y) = \frac{1}{n_u}\sum_{k=1}^{\tau} P_k(u,y)\mathbb{I}\{y(j) \in \mathcal{H}(u)\}$ where n_u is the smaller of the number of hidden songs and τ.
- MAP: average of AP over all users

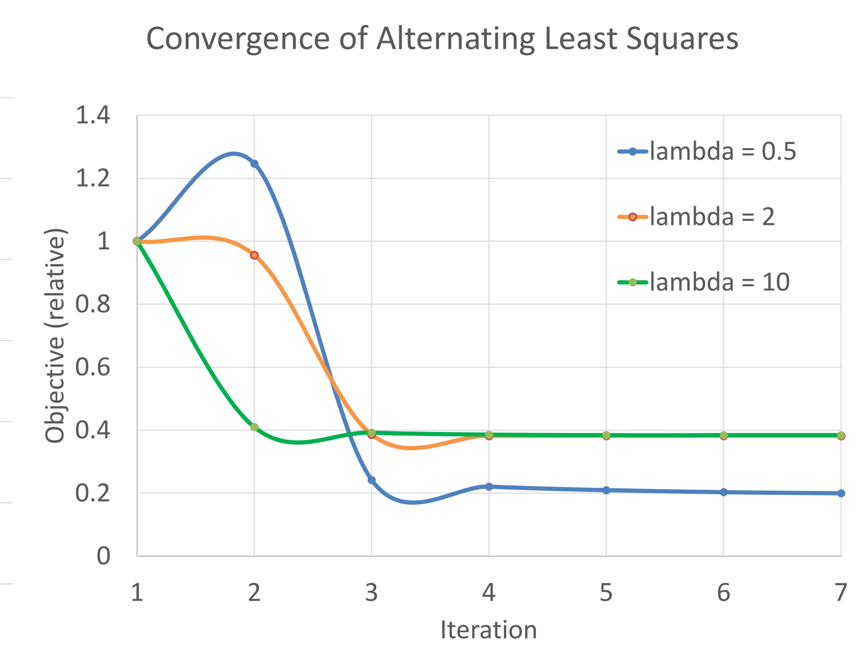
For 10K training users, 1K test users



For 50K training users, 5K test users







References

[1] Michael D Ekstrand, John T Riedl, and Joseph A Konstan. Collaborative filtering recommender systems. Foundations and Trends in Human-Computer Interaction, 4(2):81-173, 2011.
[2] Paul B Kantor, Lior Rokach, Francesco Ricci, and Bracha Shapira. Recommender systems handbook. Springer, 2011.
[3] Jure Leskovec. CS246 Course. 2014.
[4] Brian McFee, Thierry Bertin-Mahieux, Daniel PW Ellis, and Gert RG Lanckriet. The million song dataset challenge. In Proceedings of the 21st international conference companion on World Wide Web, pages 909-916. ACM, 2012.