A Latent Source Model for Online Collaborative Filtering

Guy Bresler George H. Chen Devavrat Shah

Motivation

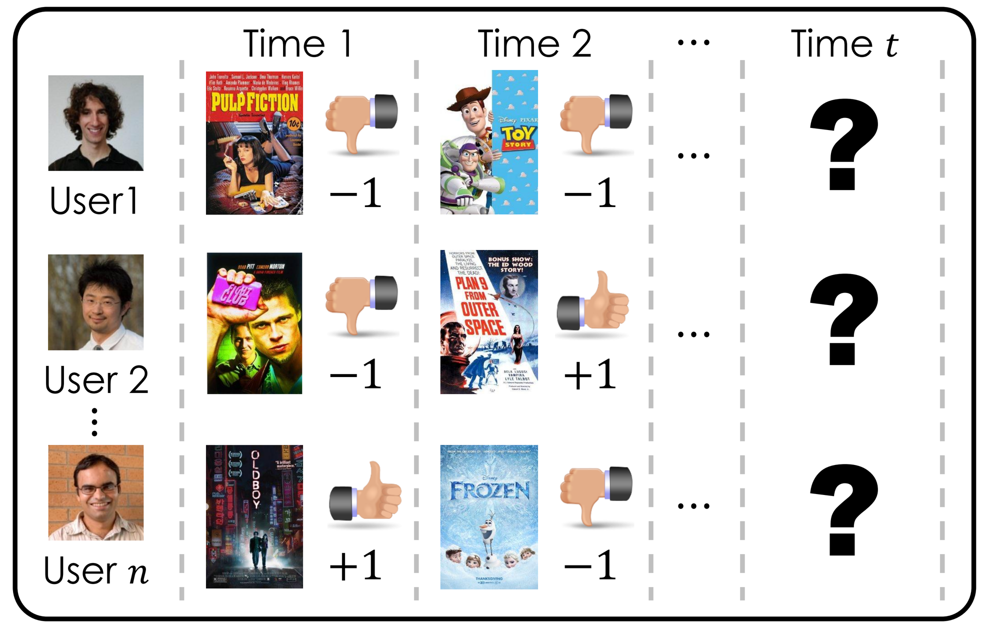
Online recommendation systems
- Recommend items to users over time
- Want to simultaneously recommend good items & learn user preferences
- Collaborative filtering widely used in practice
  → little theory justifying why it works in online setting!

Key features
- Collaborative filtering is *exploitation* → how to trade off with *exploration*?
- Can't recommend already consumed item to a use
- Structure in users makes collaboration useful

Our contributions
- Frame online recommendation as a learning problem
- Provide sufficient conditions for when a cosine-similarity collaborative filtering method achieves essentially optimal performance
  → uses two exploration types: learn about items, learn about users

Model and Problem Setup

Simple online recommendation system (n users, m items)

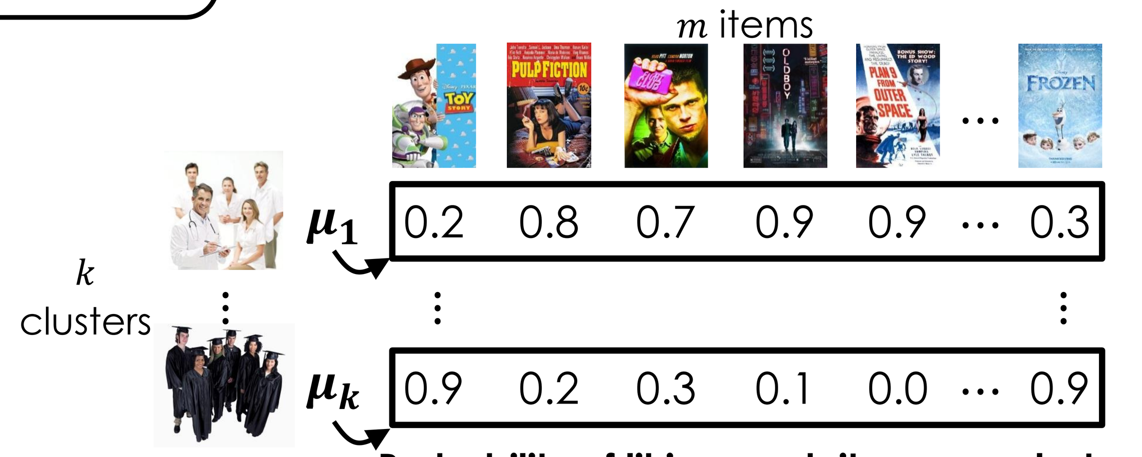


Goal: Maximize expected number of likable items recommended over time

$$r_+^{(T)} \triangleq \sum_{t=1}^{T}\sum_{u=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{\begin{array}{c}\text{item recommended}\\ \text{to user } u \text{ at time } t\\ \text{is likable}\end{array}\right\}\right]$$

How does this grow with T?

Latent source structure
- Each user belongs to one of k clusters (equally likely)
- Item is <u>likable</u> for user if the user's cluster likes the item with probability > 1/2



Probability of liking each item, per cluster

Collaborative Filtering

Exploitation: cosine-similarity nearest-neighbor recommendation

1. For user u, assign score $\hat{p}_{uj}^{(t)}$ for item j based on users' ratings up to time t:

$$\hat{p}_{uj}^{(t)} = \frac{\#\text{ neighbors of user } u \text{ who like item } j}{\#\text{ neighbors of user } u \text{ who have rated item } j}$$

Two users are neighbors ⇔ cosine similarity between their ratings $\geq \theta$

2. Recommend unconsumed item with highest score

Remarks:
- User's item score ($\hat{p}_{uj}^{(t)}$) estimates user's cluster's probability of liking the item (μ_{gj} where $g =$ user u's cluster)
- Estimate only good when enough neighbors have rated the item
  → recommendation based on item score is *exploitation*
  → need exploration!

Exploration
- Find good items:
  randomly explore items a user hasn't consumed
- Find similar users:
  ask all users to *jointly explore* common set of items

Algorithm (COLLABORATIVE-GREEDY)
Parameters: $\theta \in [0,1]$, $\alpha > 0$ sufficiently small
Select a random ordering σ of the items $[m]$
Define

$$\varepsilon_R(n) = \frac{1}{n^\alpha}, \qquad \varepsilon_J(t) = \frac{1}{t^\alpha}$$

At time t:
- W.p. $\varepsilon_R(n)$: for each user, recommend random unconsumed item **(random exploration)**
- W.p. $\varepsilon_J(t)$: for each user, recommend next unconsumed item in ordering σ **(joint exploration)**
- Else: for each user, recommend unconsumed item that maximizes $\hat{p}_{uj}^{(t)}$ **(exploitation)**

Results

Theoretical analysis
Conditions on cluster probability strings $\boldsymbol{\mu_1}, \dots, \boldsymbol{\mu_k}$:
- **Low noise.** For every cluster g and item i

$$\left|\mu_{gi} - \frac{1}{2}\right| \geq \Delta$$

Item liked w.p. close to 1/2 too ambiguous!

- **Cosine separation.** For any two different clusters g and h

$$\frac{1}{m}\langle 2\boldsymbol{\mu_g} - \mathbf{1}, 2\boldsymbol{\mu_h} - \mathbf{1}\rangle \leq 4\gamma\Delta^2$$

(E[cosine similarity] between users' ratings from clusters g and h)

Enables cosine-similarity to distinguish between clusters after enough time

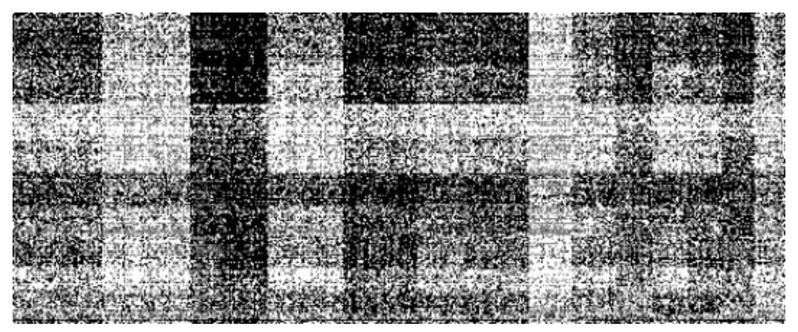
<u>Theorem</u>: Under latent source model and **low noise** and **cosine separation** conditions, with number of users $n = \Theta(km)$, after an initial learning time

$$T_{\text{learn}} = \Theta\left(\left(\frac{\log(km/\Delta)}{\Delta^4(1-\gamma)^2}\right)^{1/(1-\alpha)}\right),$$

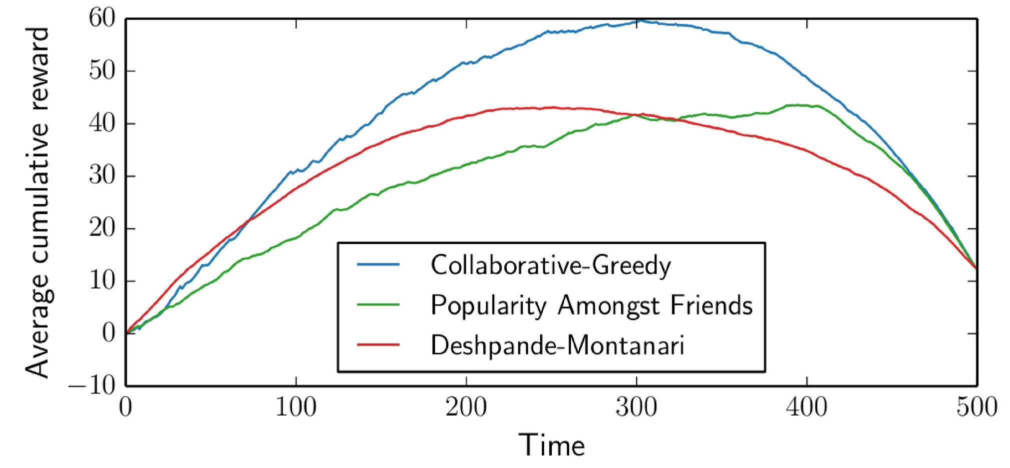
at each time step henceforth, COLLABORATIVE-GREEDY with appropriately chosen parameters recommends likable items for each user w.h.p. provided that the system hasn't exhausted the likable items for that user.

→ Fraction of likable items recommended: $\frac{r_+^{(T)}}{Tn} = \Omega\left(1 - \frac{T_{\text{learn}}}{T}\right)$
for $T_{\text{learn}} \leq T \leq \lambda m$ where $\lambda =$ minimum fraction of likable items in a cluster

Simulation results
- For dense (200 user by 500 item) subset of movielens10m & Netflix datasets, reveal entries over time to simulate online recommendation system (ratings quantized to $+1, 0, -1$)
- Look at cumulative sum of ratings averaged across users



User ratings do actually cluster!



This work was supported in part by NSF grant CNS-1161964 and by Army Research Office MURI Award W911NF-11-1-0036. GHC was supported by an NDSEG fellowship.