Recommendation under Constraints

Konstantina Christakopoulou, Jaya Kawale, Arindam Banerjee
Recommendation is Everywhere!
UNCONSTRAINED OFFLINE RECOMMENDATION
Latent Factor-based Recommenders

\[ \hat{R} = U^T V \]
Embed users-items to same low-d space
Probabilistic Matrix Factorization

\[ \hat{R} = U^T V \]

\[
\mathcal{E}_{PMF}(U, V) = \sum_{i=1}^{M} \sum_{j \in L_i} (r_{i,j} - u_i^T v_j)^2 + \lambda (\|U\|_F^2 + \|V\|_F^2)
\]

rating prediction loss \quad \text{regularization}

[Salakhutdinov and Mnih, NIPS ’07]
Bayesian Personalized Ranking

- $\Pi_{(i,k,j)} p(k >_i j | \Theta), \forall k \in L_{i,+}, j \in L_{i,-}$

- $p(k >_i j | \Theta) = \sigma(\hat{x}_{ikj}(\Theta))$

- $\hat{x}_{ikj}(U, V) = \hat{x}_{ik} - \hat{x}_{ij} = u_i^T v_k - u_i^T v_j$

Maximize the probability that $k$ ranked higher than $j$ for all pairs over users

For MF-based

$\hat{R} = U^T V$

$E_{BPR}(U, V) = \sum_{i=1}^{M} \sum_{k \in L_{i,+}} \sum_{j \in L_{i,-}} \log(1 + \exp(-u_i^T(v_k - v_j)))$

$+ \lambda(\|U\|_F^2 + \|V\|_F^2)$. 

Pairwise ranking loss

[Rendle et. al UAI '09]
Point-of-Interest Data

Every POI is associated with a \textit{latitude/longitude} coordinate

Human Mobility Patterns

If a user has visited POI j, but not nearby located POIs → more possible the user will dislike \textit{these} nearby POIs compared to far-away.
Geographical Matrix Factorization

\[ \hat{r}_{ij} = u_i^T v_j + x_i^T y_j \]

Prob. user \( i \) will visit grid \( l \)

Activity area

Check-in Data

Influence of POI \( j \) on grid \( l \)

L: grids

[Lian et al. KDD’14]
Mercator Projection: Lat/Long Coordinates $\rightarrow$ Tile Coordinates

(a) Foursquare, lat./long. coordinates  (b) Foursquare, tile coordinates  (c) Gowalla, lat./long. coordinates  (d) Gowalla, tile coordinates

Location information Scatter plots.

Pre-compute Influence matrix $Y$ via Kernel Density Estimation

\[ y_{j\ell} = \frac{1}{\sigma} \mathcal{K}\left(\frac{d(j,\ell)}{\sigma}\right) \]

Distance of POI $j$ to grid $\ell$

Fixing $Y$, learn $U$, $V$, $X$
CAPACITY CONSTRAINTS

Recommendation under Capacity Constraints. K. Christakopoulou, Jaya Kawale and A. Banerjee.
arXiv 1701.05228
Scenario 1
Recommend to most people the same Broadway show
Scenario 2
Recommend to most people the same product
Scenario 3
In busy times, e.g. Prime Time or the Oscars, if too much traffic:

Server overloaded
We are sorry, the server is overloaded at the moment. Please try again in few minutes.
Scenario 4

If we recommend same attraction park, same coffee shop, etc.
Recommendation under Capacity Constraints
A Sketch

Minimize:

Recommendation Loss + Capacity Violation Loss + Regularization

(1 − a)

a

Multi-Objective Optimization
Key Concept no.1: User Propensity to Listen

\[ p \in \mathbb{R}^M \]

\[ p_i \in [0, 1] \]

User Propensity vector

For 5 users:
- 0.2
- 0.7
- 0.3
- 0.1
- 0.6

One Possible Propensity Definition:

\[ p_i = \frac{\# \text{ times user } i \text{ followed the recommendation}}{\# \text{ user } i\text{-system interactions}} \]

Future more intricate models:
- Vary with time of day
- Vary based on who the users are with etc.
- Vary with system interaction
Key Concept no.2: Item Capacities

\[ \mathbf{c} \in \mathbb{R}^{N}_+ \]

- \( c_j > 0 \)

For 4 POIs:

- Total # of seats
- Total # of visitors allowed at the same time
- # of copies of the same item in the inventory
- Etc.
Computing Expected Usage

- If \( \hat{r}_{ij} \) 1/0, and all users follow the recommendation, total number of visitors:
  \[
  \sum_{i=1}^{M} \hat{r}_{ij}
  \]

- Every user has a probability of listening \( p_i \)
  \[
  \mathbb{E}[\text{usage}(j)] = \sum_{i=1}^{M} p_i \sigma(\hat{r}_{ij})
  \]

- General Framework
  \( \hat{r}_{ij} \) can be estimated using different models.
  \[
  \hat{r}_{ij} = \begin{cases} 
    u_i^T v_j & \text{for Cap-PMF} \\
    u_i^T v_j + x_i^T y_j & \text{for Cap-GeoMF}
  \end{cases}
  \]
Computing Expected Capacity Violation

\[
\frac{1}{N} \sum_{j=1}^{N} 1[c_j \leq \mathbf{E}[\text{usage}(j)]]
\]

Replace Ind. Function with convex increasing loss

Expected usage - capacity

\[
\Delta(c_j, \mathbf{E}[\text{usage}(j)]) = c_j - \mathbf{E}[\text{usage}(j)]
\]

\[
\ell(\Delta(c_j, \mathbf{E}[\text{usage}(j)])) = \log(1 + \exp(-\Delta(c_j, \mathbf{E}[\text{usage}(j)]))
\]
<table>
<thead>
<tr>
<th>C</th>
<th>2</th>
<th>10</th>
<th>4</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon 1]</td>
<td>![Icon 2]</td>
<td>![Icon 3]</td>
<td>![Icon 4]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>0.5</th>
<th>0.7</th>
<th>0.4</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon 5]</td>
<td>1</td>
<td>0.7</td>
<td>0.3</td>
<td>0.9</td>
</tr>
<tr>
<td>![Icon 6]</td>
<td>0.9</td>
<td>0.5</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>![Icon 7]</td>
<td>0.85</td>
<td>0.7</td>
<td>0.1</td>
<td>0.6</td>
</tr>
<tr>
<td>![Icon 8]</td>
<td>1</td>
<td>0.3</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>![Icon 9]</td>
<td>0.8</td>
<td>0.6</td>
<td>0.2</td>
<td>0.9</td>
</tr>
</tbody>
</table>

\[ \sigma(\hat{r}_{ij}) \]

\[
\frac{1}{N} \sum_{j=1}^{N} 1[c_j \leq E[usage(j)]]
\]

<table>
<thead>
<tr>
<th>Expected Usage</th>
<th>2.85</th>
<th>1.61</th>
<th>1.04</th>
<th>2.37</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Est. Capacity Violation</th>
<th>0.85</th>
<th>0</th>
<th>0</th>
<th>0.37</th>
</tr>
</thead>
</table>

0.305
Solving Cap-PMF

\[ \mathcal{E}_{\text{cap-PMF}}(U, V) = (1 - \alpha) \cdot \sum_{i=1}^{M} \sum_{j \in L_i} (r_{ij} - u_i^T v_j)^2 \]

\[ + \alpha \cdot \frac{1}{N} \sum_{j=1}^{N} \log \left( 1 + \exp \left( \sum_{i=1}^{M} p_i \sigma(u_i^T v_j) - c_j \right) \right) + \lambda(\|U\|_F^2 + \|V\|_F^2) \]

Gradient descent updates:

\[ \forall i = 1, \ldots, M, \quad u_i^{t+1} \leftarrow u_i^t - \eta \nabla_{u_i} \mathcal{E}_{\text{cap-MF}}(U^t, V^t, X^t) \]

\[ \forall j = 1, \ldots, N, \quad v_j^{t+1} \leftarrow v_j^t - \eta \nabla_{v_j} \mathcal{E}_{\text{cap-MF}}(U^{t+1}, V^t, X^t) \]

\[ \forall i = 1, \ldots, M, \quad x_i^{t+1} \leftarrow x_i^t - \eta \nabla_{x_i} \mathcal{E}_{\text{cap-GeoMF}}(U^{t+1}, V^{t+1}, X^t) \]
Other Variants

• Cap-PMF
• Cap-GeoMF
• Cap-BPR
• Cap-GeoBPR
• Etc.
Propensities – Possible Definitions

1. Propensities analogous to usage

‘actual’: \[ p_i = \frac{\# \text{ observed ratings for user } i}{\text{Total } \# \text{ items}} = \frac{|L_i|}{N} \]

- ‘median’: binning method, set \( p > \text{median} \) to max value (e.g. 0.45) else to min value (e.g. 0.01)

2. Propensities irrespective of usage

- ‘linear’: they follow linear function \( p = ax + b \), with \( a, b \): set to values so that \( p \)'s are in a range e.g. \([0, 0.6]\)
Capacities – Possible Definitions

1. Capacities analogous to usage – supply/demand
   - ‘actual’: \( c_j = \# \) users who have rated item \( j \)

2. Capacities reversely proportional to usage – items in high demand, low capacities

3. Capacities irrespective of usage
   - ‘uniform’
   - ‘linear mean’
   - ‘linear max’

---

Foursquare, capacities

Gowalla, capacities
What is the interplay between rating prediction and capacity loss?

(a) Movielens 100K

(b) Movielens 1M
What is the interplay between rating prediction and capacity loss? (2)

(c) Foursquare

(d) Gowalla
What is the interplay between pairwise ranking loss and capacity loss?

As $a$ increases, pairwise ranking loss does not necessarily increase.

Results with ‘actual’ Capacities.
Capacity Loss with capacities proportional to usage, can help reconstruct pairwise ranking!

 Movielens 100K

 Foursquare
What is the interplay between pairwise ranking loss and capacity loss?

Results with ‘reverse binning’ Capacities.

Two objectives now competitive -> expected trade-off
Sensitivity Analysis

Applicable to explicit & implicit data

Effect of surrogate loss

Effect of item capacities

Effect of user propensities
Effect on Top-N recommendations

(a) MovieLens 1M

(b) Foursquare
Comparison with a Post-Processing Baseline

Based on some recommendation algorithm, e.g. PMF/GeoMF etc.:

For every item:
Sort users based on predicted scores

Present the item to top cj users

Show top-N recommendation list per user

This will achieve 0 capacity loss, with a potential hit in recommendation accuracy
Comparison with a Post-Processing Baseline

Cap-PMF/Cap-GeoMF: opposite trend when rating prediction loss
- for capacities proportional/irrespective to usage: our framework better than the baseline
- for capacities inversely prop. to usage: CapMF worse than baseline
SPACE CONSTRAINTS: ACCURACY AT THE TOP

Collaborative Ranking with a Push at the Top, K. Christakopoulou and A. Banerjee, WWW’15.
Space Constraints

Screen size becomes smaller and smaller
• Recommendation as **Ranking**

• Recommendation as **Top Ranking**

• Analogy with **Learning to Rank**
Problem Setup
Ranking with a Push: *Height*

**Height** of a non-relevant item $x_j^-$ for user $\rho_i$:

$$H_i(x_j^-) = \sum_{k \in L_i^+} \mathbb{1}[f_i(x_k^+) \leq f_i(x_j^-)]$$

1. Want small Heights
2. Penalize larger heights: **Push**
Collaborative Ranking with a Push at the Top

- Collaborative:
  \[ f_i(x_k) = u_i^T v_k, \quad f_i(x_j) = u_i^T v_j \]

- Surrogate loss: logistic

\[
\tilde{H}_i(x_j^-) = \sum_{k \in L_i^+} \ell(f_i(x_k^+) - f_i(x_j^-)) = \sum_{k \in L_i^+} \ell(u_i^T (v_k - v_j)) = \sum_{k \in L_i^+} \log(1 + \exp(-u_i^T (v_k - v_j)))
\]
Algorithm: P-Norm Push

\[ E_{p\text{-push}}(U, V) = \sum_{i=1}^{m} \frac{\|\tilde{H}_i\|_p}{n_i} = \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j \in L_i^-} \left( \tilde{H}_i(x_j^-) \right)^p \]

\[ = \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j \in L_i^-} \left( \sum_{k \in L_i^+} \log \left( 1 + \exp(-u_i^T(v_k - v_j)) \right) \right)^p. \]

Gradient descent updates:

\[ u_{i}^{t+1} \leftarrow u_{i}^{t} - \eta \nabla_{u_i} E_{p\text{-push}}(U^t, V^t), \quad i = 1, \ldots, m, \quad (1) \]

\[ v_{h}^{t+1} \leftarrow v_{h}^{t} - \eta \nabla_{v_h} E_{p\text{-push}}(U^{t+1}, V^t), \quad h = 1, \ldots, n. \quad (2) \]
Treat Recommendation as a Ranking Problem

MovieLens 100K

MovieLens 1M

All Push CR methods outperform PMF in terms of AP@10.
Collaborative component helps

MovieLens 100K

MovieLens 1M

Collaboratively learning ranking function (Inf-Push CR) results in better AP@10 vs. constructing separate ranking model per user.
Other Works on Recommendation as Ranking

CofiRank: convex upper bound of NDCG

[Weimer et al., NIPS ‘08]

CLiMF: smooth lower bound of Mean Reciprocal Rank

[Y. Shi et al., RecSys’12]

Etc.
Time Constraints

Low latency applications

Time Critical Applications
Human Cognitive Load

Minimum Human Effort

(e.g. preference elicitation)

Work on Human Decision Making & HCI for RecSys
Conclusions - Recommendation under Constraints

- Capacity Constraints
- Size/Space Constraints
- Time Constraints
- Cognitive Constraints

And many more! E.g. objectivity on news, serendipity, novelty etc.
APPENDIX
<table>
<thead>
<tr>
<th>Dataset</th>
<th># Users</th>
<th># Items</th>
<th># Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movielens 100K</td>
<td>943</td>
<td>1,682</td>
<td>100,000</td>
</tr>
<tr>
<td>Movielens 1M</td>
<td>6,040</td>
<td>3,706</td>
<td>1,000,209</td>
</tr>
<tr>
<td>Foursquare</td>
<td>2,025</td>
<td>2,759</td>
<td>85,988</td>
</tr>
<tr>
<td>Gowalla</td>
<td>7,104</td>
<td>8,707</td>
<td>195,722</td>
</tr>
</tbody>
</table>
Metrics

\[ \text{RMSE} = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \frac{1}{|L_{i,\text{test}}|} \sum_{j \in L_{i,\text{test}}} (\hat{r}_{ij} - r_{ij})^2} \]

\[ \text{0/1 Pair. Loss} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{|L_{i}^-| \cdot |L_{i}^+|} \sum_{j \in L_{i,\text{test}}} \sum_{k \in L_{i,\text{test}}^+} 1[\hat{r}_{ij} \geq \hat{r}_{ik}] \]

\[ \text{Capacity Loss} = \frac{1}{N} \sum_{j=1}^{N} \ell \left( c_j - \sum_{i=1}^{M} p_i \sigma(\hat{r}_{ij}) \right) \]

\[ \text{AP}@k = \sum_{r=1}^{k} \frac{P@r \cdot \text{rel}(r)}{\min(k, \# \text{ relevant items})} \]