On the Effectiveness of Linear Models for One-Class Collaborative Filtering

Suvash Sedhain$^{1,2}$, Aditya Menon$^{2,1}$, Scott Sanner$^{3,1}$, Darius Brazinunas$^4$

Australian National University$^1$
NICTA$^2$
Oregon State University$^3$
Rakuten Kobo Inc$^4$
Recommender Systems

• Recommender Systems
  – Objective: Present personalized items to users

• Collaborative filtering
  – De-facto method for multiuser recommender systems
  – Find people like you and leverage their preferences
  – One-class: only observe positive feedback
Sneak Peak: Model Proposal

• Personalized user focused linear model
• Convex
• Embarrassingly parallel
  – Each user trained individually
State-of-the-art Collaborative Filtering

- Neighborhood methods
- Matrix Factorization
- SLIM (Sparse Linear Method)
Nearest Neighbors: A Matrix View

$\{\text{Jaccard, Cosine}\}$ similarity $S_I$ used in practice

• Keep only top $k$ similarities
• Simple, but learning is limited
Factorization Model

(Weighted) Matrix Factorization

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & \cdots & 0 \\
0 & 1 & 1 & 0 & \cdots & 1 \\
1 & 0 & 0 & 1 & \cdots & 1 \\
\vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\
1 & 0 & 1 & 0 & \cdots & 0
\end{bmatrix} =
\begin{bmatrix}
\vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\
\end{bmatrix}
\]

User Projection

Item Projection

\[
\begin{bmatrix}
A^T
\end{bmatrix}
\]

\[
\begin{bmatrix}
B
\end{bmatrix}
k \times n
\]

\[
\min_{A,B} \sum_{u \in U, i \in I} J_{ui}(R_{ui} - A_u^T B_i)^2 + \frac{\lambda}{2} (\|A\|_F^2 + \|B\|_F^2)
\]

\[
J_{ui} = [R_{ui} = 0] + \alpha \cdot [R_{ui} > 0]
\]

- Works well in general, but non-convex!
- Effectively trying to learn item-to-item similarities
- Not user-focused, complicated optimization
Recommender Systems Desiderata

- Learning based
- Convex objective
- User focused
- Parallelizable
Comparison of recommendation methods for OC-CF

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>U-KNN</td>
<td>(Herlocker et al., 1999)</td>
<td>×</td>
<td>NA</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>I-KNN</td>
<td>(Sarwar et al., 2001)</td>
<td>×</td>
<td>NA</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>PureSVD</td>
<td>(Cremonesi, Koren, and Turrin, 2010)</td>
<td>✓</td>
<td>✓*</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>WRMF</td>
<td>(Pan et al., 2008)</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>LogisticMF</td>
<td>(Johnson, 2014)</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>BPR</td>
<td>(Rendle et al., 2009)</td>
<td>✓</td>
<td>×</td>
<td>✓*</td>
<td>×</td>
</tr>
<tr>
<td>SLIM</td>
<td>(Ning and Karypis, 2011)</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>LRec</td>
<td>This paper</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Outline

• Problem statement
• Background
• LRec Model
• Experiments
• Results
• Summary
LRec

Recommendation for

\[
\begin{pmatrix}
1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 \\
\end{pmatrix}
\begin{pmatrix}
1 \\
0 \\
1 \\
0 \\
\end{pmatrix}
= 
\begin{pmatrix}
1 \\
0 \\
1 \\
0 \\
1 \\
\end{pmatrix}
\]

- Each item is a training instance
- Can be interpreted as learning user-user affinities
- Regularizer prevents from the trivial solution

Any loss function
- Squared
- Logistic

Recommendation
Learning a model per user

\[
\min_W \sum_{u \in U} \sum_{i \in I} \ell(y_i^{(u)} \cdot x, w^{(u)}) + \Omega(W);
\]

\[
\Omega(W) = \frac{\lambda}{2} \|W\|_F^2.
\]
Properties of LRec

• User focused
  – Recommendation as learning a model per user

• Convex objective
  – Guarantees optimal solution for the formulation

• Embarrassingly parallel
  – Each model is completely independent of other
Relationship with Existing Models

**LRec**

\[ \min_{\mathbf{W}} \sum_{u \in U} \sum_{i \in I} \ell(y^{(u)}_i, \mathbf{X}_i; \mathbf{w}^{(u)}) + \Omega(\mathbf{W}), \]

- **User focused**
- L2 penalty
- Optimization
  - L2 loss
  - Logistic Loss: Liblinear
    (dual iff #users >> #items)

**SLIM**

\[ \min_{\mathbf{W} \in \mathcal{C}} \sum_{i \in I} \sum_{u \in U} \ell(y^{(i)}_u, \mathbf{X}^{(i)}_u; \mathbf{w}^{(i)}) + \Omega(\mathbf{W}), \]

\[ \min_{\mathbf{W} \in \mathcal{C}} \| \mathbf{R} - \mathbf{R}\mathbf{W} \|_F^2 + \frac{\lambda}{2} \| \mathbf{W} \|_F^2 + \mu \| \mathbf{W} \|_1 \]

\[ \mathcal{C} = \{ \mathbf{W} \in \mathbb{R}^{n \times n} : \text{diag}(\mathbf{W}) = 0, \mathbf{W} \succeq 0 \} \]

- **Item focused**
- Elastic-net penalty + non-negativity constraints
- Optimization:
  - Coordinate descent
  - Levy et.al. relaxed the non-negativity constraints; optimization via SGD
    Truncated Gradient
Relationship with Existing Models

**LRec**
- Learns weight matrix via classification/regression problem
  - can be interpreted as learning user-user similarities

**Neighborhood models**
- Computes similarities using predefined similarity metrics (e.g., Cosine, Jaccard)
Relationship with Existing Models

**LRec**

- Learns weight matrix via classification/regression problem
  - can be interpreted as learning user-user similarities

**Matrix Factorization**

\[
\min_{\theta} \sum_{u \in U, i \in I} J_{ui} \cdot (R_{ui} - A_u^T B_i)^2 + \frac{\lambda}{2} \cdot (\|A\|_F^2 + \|B\|_F^2)
\]

If \( J_{ui} = 1 \)

\[
B = (AA^T + \lambda I)^{-1} AR
\]

**Recommendation**

\[
\hat{R} = SR
\]

Where, \( S = A^T (AA^T + \lambda I)^{-1} A \)

- Non Convex objective
- Low rank
- Embarrassingly parallel
- Convex objective
- Full rank
- Embarrassingly parallel
- Parallelism via distributed communication
Other Advantages of LRec

• Efficient hyper-parameter tuning for ranking
  – Validate on small subset of users

• Model can be fine-tuned per user
Other Advantages of LRec: Incorporating Side Information

- Can easily incorporate abundant item-side information
Outline

• Problem statement
• Background
• LRec Model
• Experiment & Results
• Summary
Dataset Description and Evaluation

- Movielens 1M (ML1M)
- Kobo
- Last FM (LASTFM)
- Million Song Dataset (MSD)

| Dataset   | m   | n   | $|R_{ui} > 0|$ |
|-----------|-----|-----|----------------|
| ML1M      | 6,038 | 3,533 | 575,281       |
| KOBO      | 38,868 | 170,394 | 89,815       |
| LASTFM    | 992  | 107,398 | 821,011      |
| MSD       | 1,019,318 | 384,546 | 48,373,586   |

- 10 random train-test split
  - 80%-20% split
  - For MSD, we evaluate on random 500 users
- Error bars => 95% confidence interval

Evaluation Metrics
- precision@k
- mean Average Precision@100
Experiment Setup

• Baselines
  – Most Popular

  – Neighborhood
    • User KNN (U-KNN)
    • Item KNN (I-KNN)

  – Matrix Factorization
    • PureSVD
    • WRMF
    • LogisticMF
    • Bayesian Personalized Ranking (BPR)

• SLIM

• LREC
  – Elastic Net Lrec + Non-Negativity (Lrec + Sq + L1+ NN)
  – Squared Loss LRec (Lrec + Sq)
  – Logistic Loss LRec (LRec)
Results

Evaluation of mAP@100

- LRec
- LRec+Sq
- LRec+Sq+L1+NN
- SLIM
- BPR
- LogisticMF
- WRMF
- PureSVD
- UI-KNN
- I-KNN
- Popularity

ML1M
LastFM
Kobo
MSD

Did not finish
Results

Precision@20 on ML1M and LastFM dataset
Results

Precision@20 on Kobo and LastFM dataset

Did not finish
Performance Evaluation

Users segmented by the number of observation

% improvement over WRMF on ML1M dataset
## Case Study

### Recommendation from WRMF vs LRec

<table>
<thead>
<tr>
<th>Preferred training movies</th>
<th>WRMF recommendations</th>
<th>LRec recommendations</th>
<th>Preferred test movies</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Day the Earth Stood Still, The</td>
<td>• Planet of the Apes</td>
<td>• Them!</td>
<td>• Blob, The</td>
</tr>
<tr>
<td>• Forbidden Planet</td>
<td>• Thing, The</td>
<td>• Godzilla (Gojira)</td>
<td>• Them!</td>
</tr>
<tr>
<td>• Kronos</td>
<td>• Night of the Living Dead</td>
<td>• Blob, The</td>
<td>• It Came from Outer Space</td>
</tr>
<tr>
<td>• Tarantula</td>
<td>• Star Trek: The Wrath of Khan</td>
<td>• 20,000 Leagues Under the Sea</td>
<td></td>
</tr>
<tr>
<td>• Thing From Another World, The</td>
<td>• Fly, The</td>
<td>• Soylent Green</td>
<td></td>
</tr>
<tr>
<td>• War of the Worlds, The</td>
<td>• Alien</td>
<td>• Village of the Damned</td>
<td></td>
</tr>
<tr>
<td>• It Came from Beneath the Sea</td>
<td>• Dark City</td>
<td>• Metropolis</td>
<td></td>
</tr>
<tr>
<td>• Invasion of the Body Snatchers</td>
<td>• Star Trek IV: The Voyage Home</td>
<td>• Quatermass and the Pit</td>
<td></td>
</tr>
<tr>
<td>• Earth Vs. the Flying Saucers</td>
<td>• 2001: A Space Odyssey</td>
<td>• It Came from Outer Space</td>
<td></td>
</tr>
<tr>
<td>• It Conquered the World</td>
<td>• Gattaca</td>
<td>• Plan 9 from Outer Space</td>
<td></td>
</tr>
</tbody>
</table>

LRec is more personalized
Summary

• LRec
  – Personalized user focused linear recommender
  – Convex objective
  – Embarrassingly parallel

• Future work
  – Further scale LRec
    • Computational
    • Memory footprint
Thanks