SLIM: Sparse Linear Methods for Top-N Recommender Systems

Xia Ning and George Karypis

Computer Science & Engineering
University of Minnesota, Minneapolis, MN
Email: {xning, karypis@cs.umn.edu}

December 14, 2011
Outline

1 Introduction
   - Top-N Recommender Systems
   - Definitions and Notations
   - The State-of-the-Art methods

2 Methods
   - Sparse Linear Methods for top-N Recommendation
   - Learning $W$ for SLIM
   - SLIM with Feature Selection

3 Materials

4 Experimental Results
   - SLIM on Binary Data
     - Top-N Recommendation Performance
     - SLIM for Long-Tail Distribution
     - SLIM Regularization Effects
   - SLIM on Rating Data

5 Conclusions
Outline

1. Introduction
   - *Top-N* Recommender Systems
   - Definitions and Notations
   - The State-of-the-Art methods

2. Methods
   - Sparse Linear Methods for *top-N* Recommendation
   - Learning $W$ for SLIM
   - SLIM with Feature Selection

3. Materials

4. Experimental Results
   - SLIM on Binary Data
     - *Top-N* Recommendation Performance
     - SLIM for Long-Tail Distribution
     - SLIM Regularization Effects
   - SLIM on Rating Data

5. Conclusions

Xia Ning and George Karypis — SLIM: Sparse Linear Methods for Top-N Recommender Systems
Top-N Recommender Systems

- Top-N recommendation
  - E-commerce: huge amounts of products
  - Recommend a short ranked list of items for users

- Top-N recommender systems
  - Neighborhood-based Collaborative Filtering (CF)
    - Item based [2]: fast to generate recommendations, low recommendation quality
  - Model-based methods [1, 3, 5]
    - Matrix Factorization (MF) models: slow to learn the models, high recommendation quality
  - SLIM: Sparse Linear Methods for Top-N Recommender Systems
    - Fast and high recommendation quality
Definitions and Notations

Table 1: Definitions and Notations

<table>
<thead>
<tr>
<th>Def</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(u_i)</td>
<td>user</td>
</tr>
<tr>
<td>(t_j)</td>
<td>item</td>
</tr>
<tr>
<td>(\mathcal{U})</td>
<td>all users (</td>
</tr>
<tr>
<td>(\mathcal{T})</td>
<td>all items (</td>
</tr>
<tr>
<td>(A)</td>
<td>user-item purchase/rating matrix, size (n \times m)</td>
</tr>
<tr>
<td>(W)</td>
<td>item-item similarity matrix/coefficient matrix</td>
</tr>
<tr>
<td>(a_i^\top)</td>
<td>The (i)-th row of (A), the purchase/rating history of (u_i) on (\mathcal{T})</td>
</tr>
<tr>
<td>(a_j)</td>
<td>The (j)-th column of (A), the purchase/rating history of (\mathcal{U}) on (t_j)</td>
</tr>
</tbody>
</table>

- Row vectors are represented by having the transpose superscript \(^\top\), otherwise by default they are column vectors.
- Use matrix/vector notations instead of user/item purchase/rating profiles.
The State-of-the-Art Methods

Item-based Collaborative Filtering (1)

- Item-based $k$-nearest-neighbor (item$k$NN) CF
  - Identify a set of similar items
  - Item-item similarity:
    - Calculated from $A$
    - Cosine similarity measure

$x_1$ $x_2$ $x_3$ ... $x_j$ ... $x_{m-1}$ $x_m$

$u_1$ $u_2$ $u_3$ ...

$u_i$ ...

$u_{n-1}$ $u_n$

$A$

$W$

Xia Ning and George Karypis  •  SLIM: Sparse Linear Methods for Top-N Recommender Systems
The State-of-the-Art Methods

Item-based Collaborative Filtering (2)

- itemkNN recommendation
  - Recommend similar items to what the user has purchased
  \[
  \hat{a}_i^T = a_i^T \times W
  \]

- Fast: sparse item neighborhood
- Low quality: no knowledge is learned

Xia Ning and George Karypis • SLIM: Sparse Linear Methods for Top-N Recommender Systems
The State-of-the-Art Methods

Matrix Factorization (1)

- **Latent factor models**
  - **Factorize** \( A \) **into low-rank user factors** \((U)\) **and item factors** \((V^T)\)
  - \( U \) and \( V^T \) represent user and item characteristics in a common latent space
  - **Formulated as an optimization problem**

\[
\text{minimize}_{U,V^T} \frac{1}{2} \| A - UV^T \|_F^2 + \frac{\beta}{2} \| U \|_F^2 + \frac{\lambda}{2} \| V^T \|_F^2
\]
The State-of-the-Art Methods

Matrix Factorization (2)

\[
\begin{align*}
\mathbf{u}^T \mathbf{v} &= \mathbf{t}_1 \mathbf{l}_1 + \mathbf{t}_2 \mathbf{l}_2 + \cdots + \mathbf{t}_k \mathbf{l}_k \\
&= \mathbf{u}^* \mathbf{t}^1 \mathbf{t}^2 \cdots \mathbf{t}^j \cdots \mathbf{t}^{m-1} \mathbf{t}^m \times \mathbf{l}_1 \mathbf{l}_2 \mathbf{l}_3 \cdots \mathbf{l}_j \cdots \mathbf{l}_{m-1} \mathbf{l}_m
\end{align*}
\]

- **MF recommendation**
  - Prediction: dot product in the latent space
    \[
    \hat{a}_{ij} = U_i^T V_j
    \]
  - Slow: dense \(U\) and \(V^T\)
  - High quality: user tastes and item properties are learned

Xia Ning and George Karypis

\*SLIM: Sparse Linear Methods for Top-N Recommender Systems
Outline

1. Introduction
   - Top-N Recommender Systems
   - Definitions and Notations
   - The State-of-the-Art methods

2. Methods
   - Sparse Linear Methods for top-N Recommendation
   - Learning $W$ for SLIM
   - SLIM with Feature Selection

3. Materials

4. Experimental Results
   - SLIM on Binary Data
     - Top-N Recommendation Performance
     - SLIM for Long-Tail Distribution
     - SLIM Regularization Effects
   - SLIM on Rating Data

5. Conclusions
Motivations:
- recommendations generated fast
- high-quality recommendations
- “have my cake and eat it too”

Key ideas:
- retain the nature of itemkNN: sparse $W$
- optimize the recommendation performance: learn $W$ from $A$
  - sparsity structures
  - coefficient values
Learning $W$ for SLIM

- The optimization problem:

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| A - AW \|_F^2 + \frac{\beta}{2} \| W \|_F^2 + \lambda \| W \|_1 \\
\text{subject to} & \quad W \geq 0 \\
& \quad \text{diag}(W) = 0,
\end{align*}$$

(1)
Learning $W$ for SLIM

- The optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|A - AW\|_F^2 + \frac{\beta}{2} \|W\|_F^2 + \lambda \|W\|_1 \\
\text{subject to} & \quad W \geq 0 \\
& \quad \text{diag}(W) = 0,
\end{align*}
\]

- Computing $W$:
  - The columns of $W$ are independent: easy to parallelize
  - The decoupled problems:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|a_j - Aw_j\|_2^2 + \frac{\beta}{2} \|w_j\|_2^2 + \lambda \|w_j\|_1 \\
\text{subject to} & \quad w_j \geq 0 \\
& \quad w_{j,i} = 0,
\end{align*}
\]
Reducing model learning time

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|a_j - Aw_j\|_2^2 + \frac{\beta}{2} \|w_j\|_2^2 + \lambda \|w_j\|_1
\end{align*}
\]

- \text{fsSLIM: SLIM with feature selection}
  - Prescribe the potential non-zero structure of \(w_j\)
  - Select a subset of columns from \(A\)

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|a_j - A'w_j\|_2^2 + \frac{\beta}{2} \|w_j\|_2^2 + \lambda \|w_j\|_1
\end{align*}
\]

Xia Ning and George Karypis • SLIM: Sparse Linear Methods for Top-N Recommender Systems
Outline

1. Introduction
   - Top-N Recommender Systems
   - Definitions and Notations
   - The State-of-the-Art methods

2. Methods
   - Sparse Linear Methods for top-N Recommendation
   - Learning $W$ for SLIM
   - SLIM with Feature Selection

3. Materials

4. Experimental Results
   - SLIM on Binary Data
     - Top-N Recommendation Performance
     - SLIM for Long-Tail Distribution
     - SLIM Regularization Effects
   - SLIM on Rating Data

5. Conclusions

Xia Ning and George Karypis

SLIM: Sparse Linear Methods for Top-N Recommender Systems
Datasets, Evaluation Methodology and Metrics

Table 2: The Datasets Used in Evaluation

<table>
<thead>
<tr>
<th>dataset</th>
<th>#users</th>
<th>#items</th>
<th>#trns</th>
<th>rsize</th>
<th>csize</th>
<th>density</th>
<th>ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccard</td>
<td>42,067</td>
<td>18,004</td>
<td>308,420</td>
<td>7.33</td>
<td>17.13</td>
<td>0.04%</td>
<td>-</td>
</tr>
<tr>
<td>ctlg2</td>
<td>22,505</td>
<td>17,096</td>
<td>1,814,072</td>
<td>80.61</td>
<td>106.11</td>
<td>0.47%</td>
<td>-</td>
</tr>
<tr>
<td>ctlg3</td>
<td>58,565</td>
<td>37,841</td>
<td>453,219</td>
<td>7.74</td>
<td>11.98</td>
<td>0.02%</td>
<td>-</td>
</tr>
<tr>
<td>ecmrc</td>
<td>6,594</td>
<td>3,972</td>
<td>50,372</td>
<td>7.64</td>
<td>12.68</td>
<td>0.19%</td>
<td>-</td>
</tr>
<tr>
<td>BX</td>
<td>3,586</td>
<td>7,602</td>
<td>84,981</td>
<td>23.70</td>
<td>11.18</td>
<td>0.31%</td>
<td>1-10</td>
</tr>
<tr>
<td>ML10M</td>
<td>69,878</td>
<td>10,677</td>
<td>10,000,054</td>
<td>143.11</td>
<td>936.60</td>
<td>1.34%</td>
<td>1-10</td>
</tr>
<tr>
<td>Netflix</td>
<td>39,884</td>
<td>8,478</td>
<td>1,256,115</td>
<td>31.49</td>
<td>148.16</td>
<td>0.37%</td>
<td>1-5</td>
</tr>
<tr>
<td>Yahoo</td>
<td>85,325</td>
<td>55,371</td>
<td>3,973,104</td>
<td>46.56</td>
<td>71.75</td>
<td>0.08%</td>
<td>1-5</td>
</tr>
</tbody>
</table>

- Datasets: 8 real datasets of 2 categories
- Evaluation methodology: Leave-One-Out cross validation
- Evaluation metrics
  - Hit Rate: \( HR = \frac{\#hits}{\#users} \)
  - Average Reciprocal Hit-Rank (ARHR) \([2]\):
    \[
    ARHR = \frac{1}{\#users} \sum_{i=1}^{\#hits} \frac{1}{p_i}
    \]
Introduction

1. Top-N Recommender Systems
2. Definitions and Notations
3. The State-of-the-Art methods

Methods

1. Sparse Linear Methods for top-N Recommendation
2. Learning $W$ for SLIM
3. SLIM with Feature Selection

Materials

Experimental Results

1. SLIM on Binary Data
   - Top-N Recommendation Performance
   - SLIM for Long-Tail Distribution
   - SLIM Regularization Effects
2. SLIM on Rating Data

Conclusions
SLIM on Binary Data

Top-N recommendation performance

Figure 1: HR comparison

Figure 2: ARHR comparison

Figure 3: learning time comparison

Figure 4: testing time comparison

Xia Ning and George Karypis • SLIM: Sparse Linear Methods for Top-N Recommender Systems
SLIM on Binary Data

SLIM for Long-Tail Distribution

Figure 5: Rating Distribution in ML10M

- SLIM outperforms the rest methods on the “long tail”.

Figure 6: HR in ML10M tail

Figure 7: ARHR in ML10M tail
SLIM on Binary Data

SLIM Recommendations for Different top-N

The performance difference between SLIM and the best of the other methods are higher for smaller values of $N$.

SLIM tends to rank most relevant items higher than the other methods.
SLIM on Binary Data

SLIM Regularization Effects

Figure 10: SLIM Regularization Effects on BX

\[
\text{minimize}_{W} \frac{1}{2} \|A - AW\|_F^2 + \frac{\beta}{2} \|W\|_F^2 + \lambda \|W\|_1
\]

- As greater $\ell_1$-norm regularization (i.e., larger $\lambda$) is applied, lower recommendation time is achieved, indicating that the learned $W$ is sparser.
- The best recommendation quality is achieved when both of the regularization parameters $\beta$ and $\lambda$ are non-zero.
- The recommendation quality changes smoothly as the regularization parameters $\beta$ and $\lambda$ change.
SLIM on Rating Data

Top-N recommendation performance

Figure 11: SLIM on Netflix

Evaluation metrics:

- per-rating Hit Rate: rHR

- All the -r methods produce higher hit rates on items with higher ratings.

- The -r methods outperform -b methods on high-rated items.

- SLIM-r consistently outperforms the other methods on items with higher ratings.
Outline

1 Introduction
   - Top-N Recommender Systems
   - Definitions and Notations
   - The State-of-the-Art methods

2 Methods
   - Sparse Linear Methods for top-N Recommendation
   - Learning $W$ for SLIM
   - SLIM with Feature Selection

3 Materials

4 Experimental Results
   - SLIM on Binary Data
     - Top-N Recommendation Performance
     - SLIM for Long-Tail Distribution
     - SLIM Regularization Effects
   - SLIM on Rating Data

5 Conclusions
Conclusions

- **SLIM**: Sparse Linear Method for *top-N* recommendations
  - The recommendation score for a new item can be calculated as an aggregation of other items
  - A sparse aggregation coefficient matrix $W$ is learned for SLIM to make the aggregation very fast
  - $W$ is learned by solving an $\ell_1$-norm and $\ell_2$-norm regularized optimization problem such that sparsity is introduced into $W$
  - Fast and efficient
P. Cremonesi, Y. Koren, and R. Turrin.
Performance of recommender algorithms on top-n recommendation tasks.

M. Deshpande and G. Karypis.
Item-based top-n recommendation algorithms.

Collaborative filtering for implicit feedback datasets.

Bpr: Bayesian personalized ranking from implicit feedback.

V. Sindhwani, S. S. Bucak, J. Hu, and A. Mojsilovic.
One-class matrix completion with low-density factorizations.

R. Tibshirani.
Regression shrinkage and selection via the lasso.
Thank You!