**Introduction**

- Top-N recommender systems: recommend ranked lists of items so as to help the users in identifying the items that best fit their personal tastes.
  - Collaborative-filtering-based algorithms [2]
  - Matrix-factorization-based algorithms [6]
  - Ranking-based algorithms [7]
- Top-N recommendation with Side Information: additional information associated with the items
  - Hybrid methods [3]
  - Matrix/tensor factorizations [4]
  - Other recommendation methods [1]
- Sparse Linear Methods with Side Information: SSLIM
  - Learn a sparse coefficient matrix for the items
  - Leverage both user-item purchase profiles and side information within a regularized optimization process
  - Sparse solution via ℓ2-norm regularization

**Sparse Linear Methods with Side Information: SSLIM**

- Item-based model: the recommendation score on an un-purchased item $t_j$ of a user $u_i$ is calculated as a sparse aggregation of items that have been purchased by $u_i$:

$$m_{ij} = m_{i}^T s_j,$$

(1)

- $m_i$: the purchase profile vector for $u_i$, on all the $n$ items
- $m_i = 1$ if $u_i$ has purchased item $t_j$ and 0 otherwise
- $s_j$: a sparse size-$n$ column vector of aggregation coefficients.
- Recommendation: for $u_i$, sort $u_i$’s non-purchased items based on their recommendation scores in decreasing order and recommending the top-$N$ items.
- Item-item aggregation coefficient matrix $S$:

$$\begin{align*}
    \text{minimize} & \quad \frac{1}{2} \| M - MS \|^2_F + \frac{\alpha}{2} \| F - FS \|^2_F + \frac{\beta}{2} \| S \|^2_F + \lambda \| S \|_1 \\
    \text{subject to} & \quad S \geq 0
\end{align*}$$

(2)

- $M = [m_1, \ldots, m_n]^T$: the user-purchase profile matrix
- $\| \cdot \|^2_F$: the matrix Frobenius norm
- $\| \cdot \|_1$: the entry-wise ℓ1-norm of $S$
- $\lambda \| S \|_1$: introduces sparsity in the solution
- $S \geq 0$: constraint: the learned $S$ represents non-negative relations between items
- $\text{diag}(S) = 0$: avoid trivial solutions (i.e., the optimal $S$ is an identical matrix) and ensure that $m_i$ is not used to compute $m_i^T$.

**Evaluation Methodology & Metrics**

- Hit Rate (HR) [2]

$$HR = \frac{\# \text{hits}}{\# \text{users}}$$

(3)

- $\# \text{users}$: the total number of users
- $\# \text{hits}$: the number of users whose item in the testing set is recommended (i.e., hit) in the size-$N$ recommendation list
- Average Reciprocal Hit-Rank (ARHR) [2]

$$\text{ARHR} = \frac{1}{\# \text{users}} \sum_{i=1}^{\# \text{users}} \frac{1}{p_i}$$

(4)

- $p_i$: the position of the item in the ranked recommendation list, if an item of a user is hit

**Datasets**

**Table 1: The Datasets Used in Evaluation**

<table>
<thead>
<tr>
<th>dataset</th>
<th>collaborative information</th>
<th>side information</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML100K</td>
<td>943</td>
<td>1622</td>
</tr>
<tr>
<td>NF</td>
<td>3,086</td>
<td>6,909</td>
</tr>
<tr>
<td>MovieLens</td>
<td>18,200</td>
<td>35,450</td>
</tr>
<tr>
<td>CWRMF</td>
<td>11,300</td>
<td>23,200</td>
</tr>
<tr>
<td>SSLIM</td>
<td>14,400</td>
<td>26,620</td>
</tr>
<tr>
<td>SLIM</td>
<td>160,626</td>
<td>311,626</td>
</tr>
<tr>
<td>SLIM1</td>
<td>16,065</td>
<td>26,065</td>
</tr>
<tr>
<td>SLIM2</td>
<td>26,065</td>
<td>466,068</td>
</tr>
<tr>
<td>SLIM</td>
<td>26,065</td>
<td>466,068</td>
</tr>
<tr>
<td>SSLIM1</td>
<td>26,065</td>
<td>466,068</td>
</tr>
<tr>
<td>SSLIM2</td>
<td>26,065</td>
<td>466,068</td>
</tr>
<tr>
<td>itemSI</td>
<td>6.8, 11.1, 0.05%</td>
<td></td>
</tr>
<tr>
<td>CWRMF</td>
<td>6.8, 11.1, 0.05%</td>
<td></td>
</tr>
<tr>
<td>Top-N</td>
<td>6.8, 11.1, 0.05%</td>
<td></td>
</tr>
</tbody>
</table>

**Results**

- **Figure 1: HR**
- **Figure 2: ARHR**
- **Figure 3: Lib: Top-N**
- **Figure 4: Density Studies**

**Acknowledgement**

This work was supported in part by NSF (IIS-0805230, OCI-1044011, and IIS-0820767), the DOE grant USDOE/DE-SC0006153 and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

**References**

- A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver.

  IEEE International Conference on Data Mining, pages 176–185, 2010.
- D. Agarwal and B-C. Chen.

  Regression-based latent factor models.
- A. Juno and M. Varhegyi.

  Similarity-based item recommendation algorithm.
- N. Xiong, S. Sra, and S. Smola.

  Learning attributes-to-feature mappings for cold-start recommendations.
- M. Debiasi and V. C. Tresp.

  Boosting-based recommendations in context.
  In Workshop on Learning to Rank for Information Retrieval.

  Evaluating recommendation algorithms: a comparative analysis.
- K. Ting and C. Liu.

  Sparse linear methods for top-N recommender systems.

- X. Ning and G. Karypis.

  Learning attribute-to-feature mappings for cold-start recommendations.

- A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver.

  Learning attributes-to-feature mappings for cold-start recommendations.

- X. Ning and G. Karypis.

  One-class collaborative filtering.

- J. Li and G. Karypis.

  DrugSE: an ensemble-based drug repurposing tool.

- N. Xiong, S. Sra, and S. Smola.

  Learning attributes-to-feature mappings for cold-start recommendations.

  In Workshop on Learning to Rank for Information Retrieval.
- K. Ting and C. Liu.

  Sparse linear methods for top-N recommender systems.

- A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver.

  Learning attributes-to-feature mappings for cold-start recommendations.