ClustKNN: A Highly Scalable Hybrid Model- & Memory-Based CF Algorithm

Ali Mamunur Rashid, Shyong K. Lam, George Karypis, and John Riedl
University of Minnesota
Problem Domain

• Collaborative filtering (CF)-based recommender systems (RS).

• Issue:
  – Scalability
Background: Why Recommender Systems?

Information overload:

- More than **1.3 million** articles!
- About **50 million** blogs!
- About **130 million** photos!
Background: Why Recommender Systems?

• One solution:
  – Recommender systems
    ▪ Tools that suggest items of interest based on
      • Users’ expressed preferences
      • Observed behaviors
      • Information about the items

▪ Collaborative Filtering
  • Recommendations based on like-minded users
Many CF Algorithms So Far...

- Most of the early ones: kNN
- View it as a special regression problem.
  - Nearly all statistical and ML approaches can be applied!

- Classification by Breese et al.(1998):

<table>
<thead>
<tr>
<th></th>
<th>Memory-based CF</th>
<th>Model-based CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplicity</td>
<td>📈</td>
<td>📊</td>
</tr>
<tr>
<td>Training cost</td>
<td>📈</td>
<td>📊</td>
</tr>
<tr>
<td>Online prediction cost</td>
<td>📊</td>
<td>📈</td>
</tr>
<tr>
<td>Adding new information</td>
<td>📈</td>
<td>📊</td>
</tr>
</tbody>
</table>
Many CF Algorithms So Far...

• Accuracy:
  – So far the main focus
    ▪ However, how much difference in accuracy users perceive?

• Does it **scale** though?
User-based $k$NN CF Algorithm

- Classic memory-based CF
- Assumption:
  - Linear relationship between two users’ preferences
    - User-similarities measured by Pearson correlation coeff.
- Works very well
  - Very good accuracy & Explainable to general users.
- Problem: Doesn’t scale!
  - $O(mn)$ online cost
**ClustKNN: Proposed Approach**

- Retain good properties of User-based kNN
- Make it to scale

Online cost: \( O(km) \approx O(m) \)
- \((k \ll m, k \ll n)\)
**ClustKNN: Proposed Approach**

- Bisecting k-means clustering
  - *Better* k-means
    - Cluster sizes are more uniform
    - Better results found in document clustering (Steinbach 2000)

- Similarity function:
  - Same in both cluster-building and CF
  - Nicely complements each other
Other Algorithms Considered

Model-based CF

SVD, pLSA

Personality Diagnosis, ClustKNN

Item-based KNN

User-based KNN

Memory-based CF
## Time-complexities

<table>
<thead>
<tr>
<th>CF algorithm</th>
<th>Offline</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>pLSA</td>
<td>$O(mn)$</td>
<td>$O(m)$</td>
</tr>
<tr>
<td>SVD</td>
<td>$O(n^2m + m^2n)$</td>
<td>$O(m)$</td>
</tr>
<tr>
<td>Personality Diagnosis</td>
<td>-</td>
<td>$O(mn)$</td>
</tr>
<tr>
<td><strong>CLUSTKNN</strong></td>
<td>$O(mn)$</td>
<td>$O(m)$</td>
</tr>
<tr>
<td>User-based KNN</td>
<td>-</td>
<td>$O(mn)$</td>
</tr>
<tr>
<td>Item-based KNN</td>
<td>-</td>
<td>$O(mn)$</td>
</tr>
</tbody>
</table>
Experiments: Datasets

- Movie recommendation data from [Movielens](https://movielens.org), helping you find the right movies.

<table>
<thead>
<tr>
<th>Property</th>
<th>ML1M</th>
<th>MLCurrent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of users</td>
<td>6,040</td>
<td>21,526</td>
</tr>
<tr>
<td>Number of movies</td>
<td>3,706</td>
<td>8,848</td>
</tr>
<tr>
<td>Number of ratings</td>
<td>10,000,209</td>
<td>29,333,690</td>
</tr>
<tr>
<td>Minimum $</td>
<td>u_i</td>
<td>, \forall i$</td>
</tr>
<tr>
<td>Average rating</td>
<td>3.58</td>
<td>3.43</td>
</tr>
<tr>
<td>Sparsity</td>
<td>95.5%</td>
<td>98.5%</td>
</tr>
</tbody>
</table>

Rating distribution:

![Rating distribution chart](chart)
Experiments: Evaluation Metrics

- **Prediction eval metrics**
  - NMAE
    - Divide MAE with Expected MAE
    - Limitation:
      - Same value of error: same treatment
        - No difference between two (pred, actual) pairs (5, 2) and (2, 5)
  - Expected Utility (EU)

- **Recommendation list eval metrics**
  - Precision-recall-F1
Evaluation Metric: EU

- Two tables:
  - A contingency table
    - Rows: predictions; columns: actual ratings
  - A utility table
    - Filled with a linear utility function:
      \[ U(\hat{R}_i, R_j) = R_j - 2|\hat{R}_i - R_j| \]
    - Penalizes false positives more than false negatives

\[ EU = \sum_{1 \leq i \leq 10 \atop 1 \leq j \leq 10} U(\hat{R}_i, R_j) P(\hat{R}_i | R_j) \]
Results

![Graph showing Expected Utility vs. # of clusters in the model](image)

![Graph showing NMAE vs. # of clusters in the model](image)
## Results: Prediction Accuracy

<table>
<thead>
<tr>
<th>CF algorithm</th>
<th>MAE</th>
<th>NMAE</th>
<th>EU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ML1M</td>
<td>MLCurrent</td>
<td>ML1M</td>
</tr>
<tr>
<td>SVD</td>
<td>0.69</td>
<td>-</td>
<td>0.43</td>
</tr>
<tr>
<td>User-based KNN</td>
<td>0.70</td>
<td>0.61</td>
<td>0.44</td>
</tr>
<tr>
<td>Item-based KNN</td>
<td>0.70</td>
<td>0.60</td>
<td>0.44</td>
</tr>
<tr>
<td>CLUSTKnn ($k=200$)</td>
<td>0.72</td>
<td>0.62</td>
<td>0.45</td>
</tr>
<tr>
<td>pLSA</td>
<td>0.72</td>
<td>0.61</td>
<td>0.45</td>
</tr>
<tr>
<td>Personality Diagnosis</td>
<td>0.77</td>
<td>0.66</td>
<td>0.48</td>
</tr>
</tbody>
</table>
## Results: Recommendation List

<table>
<thead>
<tr>
<th>CF algorithm</th>
<th>Precision top-3</th>
<th>F1 top-3</th>
<th>Precision top-10</th>
<th>F1 top-10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ML1M</td>
<td>MLCurrent</td>
<td>ML1M</td>
<td>MLCurrent</td>
</tr>
<tr>
<td>SVD</td>
<td>0.8399</td>
<td>-</td>
<td>0.379</td>
<td>-</td>
</tr>
<tr>
<td>User-based KNN</td>
<td>0.833</td>
<td>0.6693</td>
<td>0.379</td>
<td>0.4086</td>
</tr>
<tr>
<td>Item-based KNN</td>
<td>0.819</td>
<td>0.657</td>
<td>0.374</td>
<td>0.407</td>
</tr>
<tr>
<td>ClustKNN (k=200)</td>
<td>0.825</td>
<td>0.659</td>
<td>0.377</td>
<td>0.407</td>
</tr>
<tr>
<td>pLSA</td>
<td>0.817</td>
<td>0.656</td>
<td>0.375</td>
<td>0.406</td>
</tr>
<tr>
<td>Personality Diagnosis</td>
<td>0.789</td>
<td>0.622</td>
<td>0.366</td>
<td>0.391</td>
</tr>
</tbody>
</table>
ClustKNN: Discussion

- Scalable!
- Simple and explainable
- Hybrid of model- and memory-based approaches
- Great for occasionally-connected, low-storage devices!
  - Memory requirement: only $O(km + m)$!
Thanks for listening!

Questions?