Nantonac Collaborative Filtering
A Model-Based Approach

Toshihiro Kamishima and Shotaro Akaho
http://www.kamishima.net/
National Institute of Advanced Industrial Science and Technology (AIST)
RecSys 2010 @ 26-30 Sep. 2010, Barcelona, Spain
Many prediction algorithms and user interfaces are developed for recommender systems.

**BUT**

For collecting users’ preference data, almost all systems use a *rating method* or a *scoring method*.

We proposed to use a *Ranking method*.
Previous and New Contribution

Our previous contributions are…

★ proposed to use a ranking method for collecting preference data
★ developed a technique that enables to apply ranking data to existing recommendation algorithms designed for scores
★ prediction accuracies are improved by using a ranking method in comparison with a scoring method

BUT

Superiority of a ranking method is tested only for memory-based algorithms

A ranking method is also effective for model based algorithms: matrix decomposition and pLSA
Rating / Scoring Methods

**Scoring Method**
Items are evaluated by using scales with scores, s.g., a five-points-scale
The user selects “5” in a five-point scale if she prefers the item A

**Rating Method**
Items are evaluated by using ordered ratings, s.g., {good, fair, poor}
The user selects “good” if she sets a high value the item A
Ranking Method

Objects are sorted according to the degree of preference
The user prefers the item A most, and the item B least

Rank = 1 2 3
Ranks to Scores

We developed a simple technique to convert ranks in preferential orders to preferential scores based on order statistics theory.

\[ \text{expectation of ranks in a unobserved complete order} \propto \frac{\text{rank in a observed order}}{\text{(length of a observed order) } + 1} \]

Example:

itemA > itemC > itemB

rank of item A is 1

length of observed order is 3

Set the score \( \frac{1}{4} = \frac{1}{(3 + 1)} \) to the item A
Assumption

consisting of all possible objects and unobserved

**unobserved**

complete order

A < B < C < D < E

1 2 3 4 5

consisting of sub-sampled objects and observed

**observed**

sample order

A < B < D

1 2 3

uniformly at random

miss

C

E

expectation of rank in a complete order

observed rank

3
Interface

WWW Interface for asking user preference by ranking method

1. show 10 items to the user
2. the user specify all the rank of each items
3. press “submit” button
4. if error (ex. the same ranks are assigned to the two items) is detected, the system request to re-input

name of sushi  Specify Ranks
**Memory-Based Method**

Groups like memory-based method
- ranking method + default voting
- scoring method + default voting + standardization (min-max range)
- scoring method + default voting + rank correlation
Matrix Decomposition Model

The user x’s score to the item y is estimated by the following Eq.

\[ \hat{s}_{xy} = b + c_x + d_y + u_y^\top \left( v_x + \frac{1}{\sqrt{|Y_x|}} \sum_{y' \in Y_x} w_{y'} \right) + u_y \]

Parameters are estimated by minimizing the loss function:

\[ \text{loss}(\mathcal{D}; \Theta) = \sum_{(x,y,s) \in \mathcal{D}} (s_{xy} - \hat{s}_{xy})^2 + \lambda \text{[reg. term]} \]
Memory-based method: Matrix decomposition

- ranking method + matrix decomposition
- scoring method + matrix decomposition
Hoffman’s pLSA model is modified so as to deal with real scores.

Parameters are estimated by maximizing the likelihood function:

\[
\mathcal{L}(\mathcal{D}; \Theta) = \sum_{(x,y,s) \in \mathcal{D}} \log \sum_z \Pr[z] \Pr[x|z] \Pr[y|z] \mathcal{N}(s; \mu_z, \sigma_z^2)
\]
pLSA-like Model

Memory-based method: Matrix decomposition
• ranking method + pLSA-like model
• scoring method + pLSA-like model
Why Ranking performed better?

The degree of true preference cannot be observed directly.

Each user uses one’s own mapping from the degree to rating score.

Ex: The degree of preference on X lies in interval 2 of user A.

User A replies rating score 2.
Why Ranking performed better?

We now want to induce the true degree of preference

The true mapping to rating scores is unknown

A common idealized mapping scale is of necessity used

The induced degrees of preferences might not be true

Ex: The true degrees of X, Y, and Z are changed to X’, Y’, and Z’, respectively

<table>
<thead>
<tr>
<th>User A</th>
<th>X → 2</th>
<th>Y → 3</th>
<th>Z → 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>User B</td>
<td>X → 3</td>
<td>Y → 2</td>
<td>Z → 5</td>
</tr>
</tbody>
</table>
Why Ranking performed better?

In a ranking method, the degrees of preferences are relatively specified.

We don’t need to use a unsafe groundless mapping between the degrees of preference and observed rating scores.
Merits and Demerits of Ranking Methods

**Merits**

- High consistency of preferences between and within users

**Demerits**

- Less algorithms for analysis are available
  - Develop new algorithms
- Difficult to rate many items at the same time
  - Subsets of items are sorted multiple times
- Lack of absolute evaluation
Ranking Many Objects

many objects

sampling small sets of objects

Sort all objects AT THE SAME TIME

No!

Iterate sampling and Sorting

OK!
Relevance Feedback
[Joachims 02, Radlinski+ 05]

Leaning from relevance feedback is a typical absolute ranking task.

The user scans this list from the top, and selected the third document C.
The user checked the documents A and B, but these are not selected.
This user’s behavior implies relevance feedbacks: C>A and C>B.

Object ranking methods can be used to update document’s relevance based on these feedbacks.
The word *nantonac* originates from a Japanese word, *nantonaku*, which means just somehow.

For example, in Japanese, if I say “I *nantonaku* understand something,” I am saying that I cannot specifically explain why I understand it, but that I somehow do.

Order responses allow a more vague and intuitive expression of users' preferences, so we have named this method the *nantonac collaborative filtering*. 

Our Related Publications

- T. Kamishima, "Nantonac Collaborative Filtering: Recommendation Based on Order Responses", Proc. of The 9th Int’l Conf. on Knowledge Discovery and Data Mining (KDD 2003)


Our SUSHI data sets are available at

http://www.kamishima.net/sushi/