

New Recommendation Techniques for Multicriteria Rating Systems

The paper by Gediminas Adomavicius
and YoungOk Kwon

Presented by Gytis Tumas
and Gregory Osunde

14/11/2007

Roadmap



- Typical Recommendation techniques
- Problem!
- Solution: Multicriteria techniques
- Experimental results

Recommender Systems

- Help consumers to avoid information overload
- Make suggestions which information is most relevant
- Live example: Yahoo! Movies

Recommendations For You Receive Recommendations by Email

Movies in Theaters: 94089

 <p>The Kingdom (R) Showtimes & Tickets Add to My Lists Yahoo! Users: B 5858 ratings The Critics: C+ 12 reviews</p> <hr/> <p><input type="checkbox"/> Don't Recommend Again <input checked="" type="checkbox"/> Seen It? Rate It!</p>	 <p>The Darjeeling Limited (R) Showtimes & Tickets Add to My Lists Yahoo! Users: B 1166 ratings The Critics: B- 13 reviews</p> <hr/> <p><input type="checkbox"/> Don't Recommend Again <input checked="" type="checkbox"/> Seen It? Rate It!</p>
---	--

[See All Recommendations](#)

Classification of recommender system

- Content based approach
- Collaborative filtering
- Hybrid

Content based

Based on what the user like and select in the past. R: USERS * ITEMS -> Ro(initial rating)

- If you watched harry potter and the “sorcerer's story”



- Today the recommender system suggests for me:
 - Harry Potter and the Chamber of Secrets
 - Polar Express

Collaborative filtering

based on users with similar taste prefer in the past.

- Identify the set of user Ratings
- The user more similar to the target user are Identified

Hybrid

□ Try to address the **shortcomings of both content-based** and collaborative-based approaches, and produce recommendations using a combination of those techniques

There is a **large variability on these hybrid methods** – there is no standard hybrid method

Classification of recommender system based on algorithmic approach

➤ Memory based

make calculation based on user stored data or previous activities.

➤ Model based

uses some machine learning from previous stored data. It train the model with past activities.

Typical Recommender Systems

- One of the Goals: estimate the ratings of unrated items on the basis of given ratings.
- Single-rating memory-based collaborative-filtering technique:

$$R(u, i) = z \sum_{u' \in N(u)} \text{sim}(u, u') \cdot R(u', i)$$

$$R(u, i) = \overline{R(u)} + z \sum_{u' \in N(u)} \text{sim}(u, u') \cdot \left(R(u', i) - \overline{R(u')} \right)$$

Collaborative-based filtering

- A simple example:

	<i>i</i>		
u_1	?		
u_5	4		
u_8	3		
u_9	5		

$\text{sim}(u_1, u_5) = 0,5$
 $\text{sim}(u_1, u_8) = 0,5$
 $\text{sim}(u_1, u_9) = 0,8$
 $\overline{R}(u_1) = 3,2$
 $\overline{R}(u_5) = 4$
 $\overline{R}(u_8) = 3,5$
 $\overline{R}(u_9) = 3$

$$R(u, i) = \overline{R}(u) + z \sum_{u' \in N(u)} \text{sim}(u, u') \cdot (R(u', i) - \overline{R}(u'))$$

$$R(u_1, i) = 3,2 + 1/(0,5 + 0,5 + 0,8) * ((0,5 * 0) + (0,5 * -0,5) + (0,8 * 2))$$

$$R(u_1, i) = 3,2 + 0,75 = \mathbf{3,95}$$

Another example

	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
Target user User u_1	5	7	5	7	
Users most similar to the target user	User u_2	5	7	5	7
	User u_3	5	7	5	7
User u_4	6	6	6	6	5
User u_5	6	6	6	6	5

Collaborative filtering in a **single-rating** setting

Multicriteria Ratings

- Example of rating a movie - how much did I like...
 1. Story
 2. Acting
 3. Direction
 4. Visuals
- Additional information can potentially increase the recommendation accuracy!

Multicriteria setting

	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5	
Target user User u_1	5 _{2,2,8,8}	7 _{5,5,9,9}	5 _{2,2,8,8}	7 _{5,5,9,9}	?	
Users most similar to the target user	User u_2	5 _{8,8,2,2}	7 _{9,9,5,5}	5 _{8,8,2,2}	7 _{9,9,5,5}	
	User u_3	5 _{8,8,2,2}	7 _{9,9,5,5}	5 _{8,8,2,2}	7 _{9,9,5,5}	
	User u_4	6 _{3,3,9,9}	6 _{4,4,8,8}	6 _{3,3,9,9}	6 _{4,4,8,8}	5
	User u_5	6 _{3,3,9,9}	6 _{4,4,8,8}	6 _{3,3,9,9}	6 _{4,4,8,8}	5

Rating to be predicted

Ratings to be used in prediction

5

Collaborative filtering in a **multicriteria** setting, where the overall Rating for each movie i is a simple average of four rating criteria: story, acting, direction and visuals.

The problem

- The recommendation of the single rating systems was worse than the one of multicriteria.
- The nearest neighbours were misidentified – there was not enough additional information
- **The more accurately the system determines who the “true peers” of a user are, the more accurate the rating prediction should be.**

Two approaches

- Similarity-based approach
- Aggregation-function-based approach

Two approaches

- **Similarity-based approach**
- *Aggregation-function-based approach*

Similarity-based approach

- This approach extends the traditional single-criteria memory-based collaborative-filtering algorithm.
- Having more info (multicriteria data) we can improve the computation of the similarity between two users
- There exist two ways:
 - Aggregating traditional similarities from individual criteria
 - Calculating similarity using multidimensional distance metrics

Aggregating traditional similarities from individual criteria

- ❖ This approach can use any of the standard based similarity such as Cosine-based to calculate the similarity between users or items on the bases of individual criterion.
- Assume that the rating of user u on item i consist r_0 - the overall rating – and k multi criteria ratings r_1, \dots, r_k . $R(u, i) = (r_0, r_1, \dots, r_k)$.
- We can obtain $k + 1$ different similarity estimations by using **some standard metric** to measure similarity between users u and u' .

Calculating the overall similarity

We compute the overall similarity by aggregating the individual similarities in one of the following 2 ways:

Average similarity

$$\text{sim}_{\text{avg}}(u, u') = \frac{1}{k+1} \sum_{i=0}^k \text{sim}_i(u, u')$$

Worst case similarity

$$\text{sim}_{\text{min}}(u, u') = \min_{i=0, \dots, k} \text{sim}_i(u, u')$$

Similarity using multidimensional distance metrics

- 3 steps:
 - Find the distance between 2 users' ratings **for the same item.**
 - Find the distance between 2 users' for all items (the overall distance)
 - Transform the distance into the similarity

Finding the distance between 2 users' ratings **for the same item**

- **Manhattan distance**

k stand for the number of criteria (# of criterion here is 4)

r	7	9,9,5,5
r'	5	2,2,8,8

$$\sum_{i=0}^k |r_i - r'_i|$$

$$(2+7 + 7 + 3 + 3) = 22$$

Finding the distance between 2 users' ratings for the same item

EUCLIDEAN DISTANCE:

The straight line distance between two points. This approach is used to calculate the similarities between users based on their criteria

$$\begin{array}{l} 7 \\ 9, 9, 5, 5 \\ 5 \\ 2, 2, 8, 8 \end{array}$$

$$\sqrt{\sum_{i=0}^k |r_i - r'_i|^2}$$

$$(2^2 + 7^2 + 7^2 + 3^2 + 3^2) = 120$$

$$(4 + 49 + 49 + 9 + 9) = 120$$

$$\text{Sqrt} (120) = 10.95$$

Finding the distance between 2 users' ratings for the same item

■ CHEBYSHEV DISTANCE:

- Chebyshev distance is also called Maximum value distance. It examines the *absolute magnitude of the differences* between coordinates of a pair of objects.

$$\begin{array}{l} 7 \\ 9, 9, 5, 5 \\ 5 \\ 2, 2, 8, 8 \end{array} \max_{i=0, \dots, k} |r_i - r'_i|$$

$$\text{Max} = (7-5, 9-2, 9-2, 5-8, 5-8)$$

$$(2+7 +7 + 3 +3) = 7$$

The overall distance B/W two users

Common
item
Rating
of u and
 u'

$$d_{\text{user}}(u, u') = \frac{1}{|I(u, u')|} \sum_{i \in I(u, u')} d_{\text{rating}}(R(u, i), R(u', i))$$

The overall distance between two user u and u' is the average distance between their ratings for all their common item

The users have Identical similarity (1) if the distance is 0. If Distance is larger similarity will approach 0.

$$\text{sim}(u, u') = \frac{1}{1 + d_{\text{user}}(u, u')}$$

Collaborative filtering operate on metric of user similarity rather than user distance, the above metric is a simple transformation of the two metric as an inverse.

Two approaches

- Similarity-based approach
- **Aggregation-function-based approach**

Aggregation-function-based approach

- Intuition: the **overall rating** is not just another rating, independent from others, it **has a relationship with the multicriteria ratings**.

- $5_{8,8,2,10}$

$$r_0 = f(r_1, \dots, r_k)$$

- Not limited to any specific recommendation algorithm

3 steps

- To rate the estimation we have to:
 1. Predict multicriteria ratings
 2. Learn the aggregation function
 3. Predict the overall rating

3 steps

- To rate the estimation we have to:
 1. **Predict multicriteria ratings**
 2. Learn the aggregation function
 3. Predict the overall rating

Predict multicriteria ratings

- Decompose:

From the k -dimensional multicriteria rating space
Into k single-rating recommendation problems

- Instead of:

$R: Users \times Items \rightarrow R_0 \times R_1 \times \dots \times R_k$

We will have to deal with k single-rating problems

$R: Users \times Items \rightarrow R_i$ (where $i=1, \dots, k$)

Predict multicriteria ratings

- More flexibility! 😊
- Not like in the Similarity-based approach, we are not constrained to the collaborative technique – we are free to use **any existing single-rating recommendation technique**.

3 steps

- To rate the estimation we have to:
 1. Predict multicriteria ratings
 2. **Learn the aggregation function**
 3. Predict the overall rating

Learn the aggregation function

- This step aims to estimate the **relationship f** between the **overall rating** and underlying **multicriteria ratings**, such that

$$r_0 = f(r_1, \dots, r_k)$$

- Having this function f, we can predict the overall rating for each item for each user.

Several ways to obtain the function

- Domain expertise
 - On the basis of prior experience and domain knowledge, a domain expert can suggest an appropriate aggregation function.
- Statistical techniques
 - A linear regression example:
$$r_0 = w_1 r_1 + \dots + w_k r_k + c$$
- Machine learning techniques
 - Usage of sophisticated computational learning techniques, e.g. artificial neural networks.

Scopes of the aggregation function

- Aggregation functions can be classified for different learning techniques.
- Beside that aggregation functions can have 3 different scopes as well (no matter which learning technique was used):
 - Total
 - User-based
 - Item-based

Scopes of the aggregation function

- If a function is used to predict all unknown ratings – it is a *total aggregation function*.
- Imagine that the user u usually gives greater weight for to the “story” component of all movies, while the user u' – to the “visuals”.
In this case the user u would benefit from having his own *user-based aggregation function* f_u

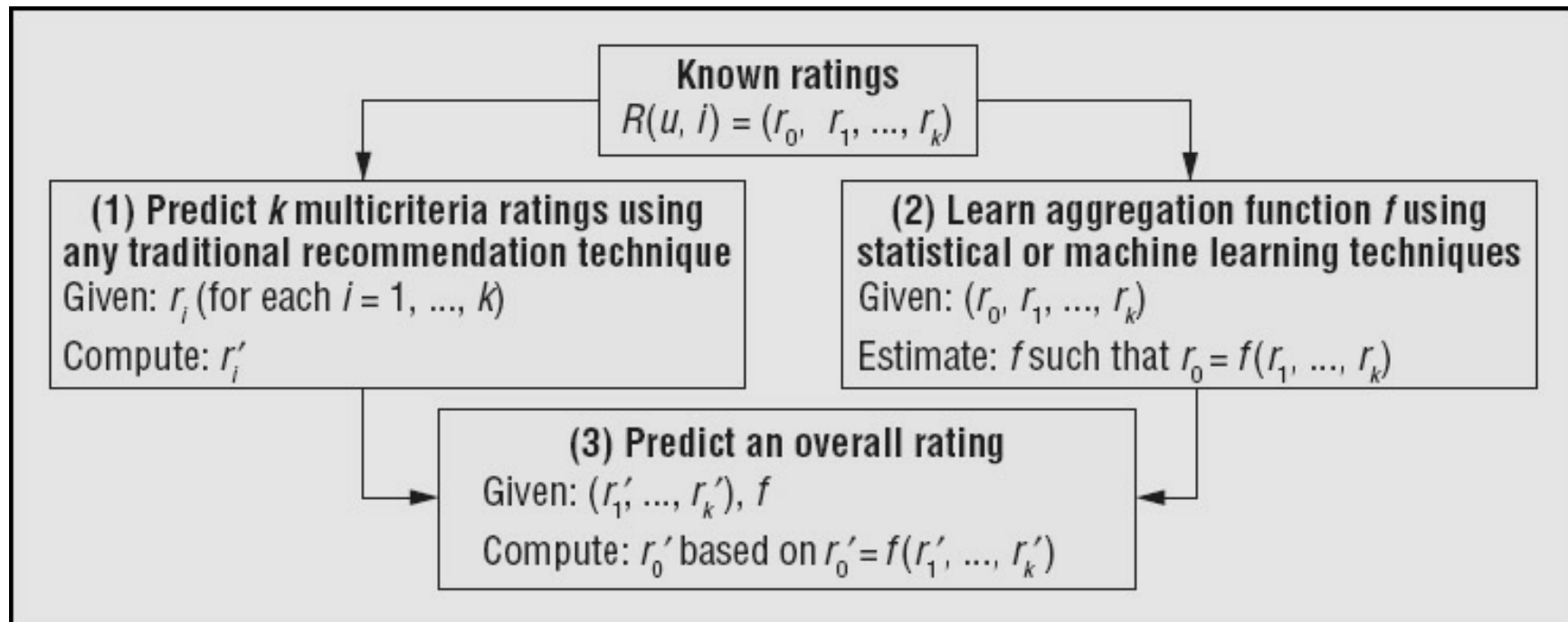
3 steps

- To rate the estimation we have to:
 1. Predict multicriteria ratings
 2. Learn the aggregation function
 3. **Predict the overall rating**

Predict the overall rating

- Finally, we compute the unknown overall rating by using the **multicriteria ratings** estimated in step1 and **function f** estimated in step2

$$r_0 = f(r_1, \dots, r_k)$$



Experimental results

- ❑ The result was evaluated from all the details collected from the yahoo movie rating.
- ❑ Four criteria was used (story, acting, direction, and visuals)
- ❑ Grading scale was from 1 to 13 (represent from A+ to F)
- ❑ Each user rate at least 10 movies and each movie has at least 10 users ratings

Computation of Sparsity

Sparsity is the proportion of known ratings over all the possible ratings

Sparsity = Number of known-ratings / Number of all-possible-ratings

- ❑ 155 movie users Rating.
- ❑ 50 movies
- ❑ 2,216 known rating in total
- ❑ The data set sparsity is 28.6 percent (28.6 percent rating known)

$$\text{Sparsity} = 2\,216 / (155 * 50) = 0.285806452$$

Precision and Recall

- ❑ **Precision is the ratio of relevant items selected by the recommender to the number of items selected ($\# \text{Relevant Selected} / \# \text{Selected}$)**
- ❑ **Recall is the ratio of relevant items selected to the number of relevant ($\# \text{RS} / \# \text{R}$)**
- ❖ **Precision metric will be used to represent the percentage of truly high overall rating among the one that the system recommend will be the relevant item for each users**
- ❖ **Precision compute the frequency with which the system correctly predict the N most relevant item.**
- ❖ **High rank fail above 10.5 (A+, A, A-) while ranking less than 10.5 are low rank 35.6 percent of the overall rating were above 10.5 and therefore 35.6 was used as the threshold.**

Experimental Result

The multi criteria techniques outperformed the single criteria approach (Standard CF or baseline)

Recommendation approach:		Precision in top 3 (%)	Precision in top 5 (%)	Precision in top 7 (%)
user-based CF				
Neighborhood size: all users	standard CF	70.7*	68.7	69.0
	cos-min	70.7**	68.8	69.1
	Chebyshev	74.5†	70.3	70.5
	total-reg	71.5	70.9	70.4
	movie-reg95	71.8	74.0‡	75.3
Neighborhood size: three users	standard CF	64.9	64.9	66.3
	cos-min	67.1	67.1	67.8
	Chebyshev	66.2	65.5	64.6
	total-reg	65.2	66.6	66.5
	movie-reg95	69.0	70.7	72.2

Single rating
CF baseline

The performance of the recommender system can be evaluated by using Decision support and root mean square error statistical model.

Move based was used against User based:- total reg performed best

Increasing the sparsity

- Experiments with a new data set's sparsity of 19,6 percent.
- Results of that: most multicriteria recommendation algorithms outperformed the baseline approach by 5 to 10 percent.

Conclusion

- The performance of a specific technique is highly domain-dependent.
- We don't expect the proposed techniques to outperform traditional single-criteria techniques in all domains where multicriteria information exists.

Summary

- General characteristics of Recommender Systems
- MULTICRITERIA
 - Similarity-based approach
 - Aggregating traditional similarities for individual criteria
 - Calculating similarity using multidimensional distance metrics
 - Aggregation-function-based approach
(Main idea – overall rating not just another rating)
 - STEP 1: Predict multicriteria ratings
 - STEP 2: Learn the aggregation function
 - STEP 3: Predict overall ratings
- Experimental results

THE END

Formulas (1)

$$z = \frac{1}{\sum_{u' \in N(u)} |\text{sim}(u, u')|}$$

$$\text{sim}(u, u') = \frac{\left(\sum_{i \in I(u, u')} R(u, i) R(u', i) \right)}{\left(\sqrt{\sum_{i \in I(u, u')} R(u, i)^2} \sqrt{\sum_{i \in I(u, u')} R(u', i)^2} \right)}$$