Extended Bayesian Model for Multi-criteria Recommender System

Pannawit Samatthiyadikun[1,a)] Atsuhiro Takasu[2,1,b)]

Abstract: We have proposed multi-criteria (MC) recommender system by using a latent probabilistic model. In this model, users and items are mapped into small number of groups, and preference is represented based on the group instead of indivisual user. In other words, features of users and items are represented by probability distributions over latent topics. When predicting rating scores, we need to aggregate features into predicted rating score. This paper compares two ways to aggregate features for predicting rating score of unrated items in MC recommendation.

Keywords: recommender system, bayesian, model, regression

1. Introduction

The information has been generated in many ways such as music, movie, advertising, brochure, and anything else. Because of growth rate of these information is rapidly rise up, the information overload problem are popped out. Many researchers have noticed it and trying to find the best way that able to handle this problem. There are so many methods that could solve this problem, among them, recommendation technology is the one that many researchers are interested in. The recommender systems extract each user ' s preferences from their information retrieval, utilization behavior from their past activities, and select the information that matches each user ' s preference.

Generally, recommender system capture the informations such as users' preference for each item, and item's content features. There are 2 main techniques which usually are implemented, the first one is collaborative filtering (CF) technique. It will search for the top-*k* nearest neighbor users who have some similar preferences, behavior or activities to the target user, and present some items which are obtained high rating scores from those neighbors. The another one is content-based filtering, the systems which adopted this technique will extract user's behavior or previous activities and use them for recommending the items to user. The one of these techniques might suits for some circumstance while the other one is not. For merging the advantages of introduced techniques, the researchers have tried to purpose a hybrid recommendation techniques. In more detail, this technique utilizes both the users' preferences and the items' features. Many researchers have reported that hybrid systems can improve recommendation accuracy.

The main topic for proposed recommender systems is its accuracy. There are so many ways that can be implemented. The one of these is using multi-criteria rating (MC) [3] which separately shows the users' preferences on each items aspect. For example, movies can be evaluated depending on both the story and the actors in them. When choosing a movie, some users are mainly influenced by its story, whereas others are influenced by the actors in it. By considering multiple aspects, the systems can obtain higher recommendation accuracy [4], [5].

Some recommender systems required a large number of ratings which will be extracted into users' preferences and items' features. Anyway, number of this kind of data is limited, and might not enough for extraction. Therefore, the other important problem for recommender system is how to extract features from sparse data. To solve this problem, machine learning techniques, such as latent probabilistic models and matrix factorization (MF) have been applied.

Our recent model, Bayesian-based multi-criteria recommender system, is the model which applied the both Flexible Mixture Model (FMM) [2], and latent Dirichlet allocation (LDA) [6] concept. By using the latent topic space via a probabilistic model, the model parameters are learned based on Bayesian estimation and has an ability to scale down the large dimensional sparse data. It showed that it improve the recommendation performance compared with the extended FMM with multi-criteria [7].

However, the MAE evaluated result does not seem so well. Therefore we need to improve our model for the better performance on MAE. The main contribution of paper are:

- extending Bayesian model by implementing with any function that we expect to improve the model performance when evaluating with MAE, and
- discussing the result provided by a new method against the original one.

1 The Graduated University for Advanced Study, Chiyoda, Tokyo 101–8430, Japan
2 National Institute of Informatics, Chiyoda, Tokyo 101–8430, Japan
a) s-pannawit@nii.ac.jp
b) takasu@nii.ac.jp

2. Problem Definition and Related Works

2.1 Multicriteria Rating Recommender System

In the most of recommender system, we can know how much user satisfy each item by observing the overall rating. Multi-criteria recommender system required additional criteria ratings as user's aspect for evaluate each item in addition to an overall score.

We have found that overall score is related to the other criteria given by each user in the same way; although, the item ratings for each criterion given by each user are different. Therefore knowing user's preference in multiple aspects; show us a better understanding of user's behavior.

Suppose that U and V are the set of users and items respectively. $C = \{c_0, c_1, \ldots, c_m\}$ is a set of criteria where c_0 and $c_i (1 \le i \le m)$ referred to overall score and score in each criterion. We also denote the preferences given by a user u for an item v as a vector; where each element referred to each criterion in the system:

$$\mathbf{r}_{uv} \equiv (r_{uvi})_{0 \le i \le m} \tag{1}$$

In our works, we use the Yahoo! Movies dataset which contains four criteria: story, acting, directing, and visual effects. On each criterion, a user u provides a rating score ranging from 1 to 13 for an item v; as represented by vector $(r_{uv0}, r_{uv1}, \ldots, r_{uvm})$.

Because each user gives the rating scores a few item, let P denotes a set of pairs of any user $u \in U$ and rated item $v \in V$ by user u. We denote the set of rating scores or training set as:

$$X \equiv \{(u, v, \mathbf{r}_{uv})\}_{(u,v) \in P} \tag{2}$$

After that, a system will use a set Y to filter out some items that might be preferred by each user. The main problem of MC recommender system is to find out how much user might like unrated items in each criterion.

2.2 Model-Based System

Generally, MC recommender systems have two steps for overall rating prediction as follows:

(1) predicting rating score for each criterion, then
(2) using predicted score in step 1 for predicting overall rating score.

The first step can be done by implementing any single-recommendation method to each criterion. In the second step, we apply an aggregate function: $r_0 = f(r_0, r_1, \ldots, r_m)$ for obtaining an overall rating. There are so many aggregate function can be use, for example: Multiple-linear regression.

2.3 Probabilistic Latent Model

We developed a probabilistic latent model for multi-criteria recommender system [1]. Figure 2.3 shows a graphical model of bayesian latent model. It is designed to calculate the probability that user u gives rating score on criterion r_c to an item v. Let us define some additional notation, T_u and T_v are the set of latent topics for a user u and an item v respectively, which represent user and item group respectively. The tuple y We define the likelihood of the complete-data rating score in this model by:

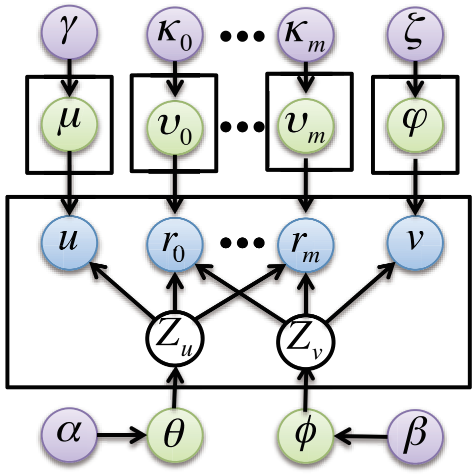

Fig. 1 Graphical model of Bayesian multi-criteria

$$P(u, v, \mathbf{r}, t_u, t_v) = P(t_u)P(t_v)P(u|t_u)P(v|t_v) \prod_{c=0}^{m} P(r_c|t_u, t_v) \tag{3}$$

The model generates a set of single rating scores using the following process:

(1) draw $\theta \sim \mathcal{DIR}(\alpha)$ where α is a vector $(\alpha_{t_u})_{t_u \in T_u}$
(2) draw $\phi \sim \mathcal{DIR}(\beta)$ where β is a vector $(\beta_{t_v})_{t_v \in T_v}$
(3) for each $t_u \in T_u$, draw $\mu_{t_u} \sim \mathcal{DIR}(\gamma)$ where γ is a vector $(\gamma_u)_{u \in U}$
(4) for each $t_v \in T_v$, draw $\varphi_{t_v} \sim \mathcal{DIR}(\zeta)$ where ζ is a vector $(\zeta_v)_{v \in V}$
(5) for each $c(0 \le c \le m), t_u \in T_u$, and $t_v \in T_v$, draw $\upsilon_{ct_ut_v} \sim \mathcal{DIR}(\kappa_\mathbf{c})$ where κ_c is a vector $(\kappa_{cr})_{r \in R}$
(6) $P(z_u) \sim Multi(\theta)$, where θ is a vector $(\theta_{t_u})_{t_u \in T_u}$
(7) $P(z_v) \sim Multi(\phi)$, where ϕ is a vector $(\phi_{t_v})_{t_v \in T_v}$
(8) for each $t_u \in T_u, P(u|t_u) \sim Multi(\mu_{t_u})$, where μ_{t_u} is a vector $(\mu_{t_u u})_{u \in U}$
(9) for each $t_v \in T_v, P(v|t_v) \sim Multi(\varphi_{t_v})$, where φ_{t_u} is a vector $(\varphi_{t_v v})_{v \in V}$
(10) for each $c(0 \le c \le m), t_u \in T_u$ and $t_v \in T_v, P(r_c|t_u, t_v) \sim Multi(\upsilon_{ct_ut_v})$, where $\upsilon_{ct_ut_v}$ is a vector $(\upsilon_{ct_ut_v r})_{r \in R}$

where the multinomial and Dirichlet distribution's parameters can be estimated by using Gibbs sampling, and Minka's fixed point iteration.

3. Preference Prediction

3.1 Bayesian Estimation

This estimation method is widely used with any probabilistic graphical model by using expectation value of rating of item v given by user u in c-th criterion (r_{uvc}), which can be computed as follows:

$$r_{uvc} = \sum_{r_c} P(r_c|u, v) r_c = \frac{\sum_{r_c, t_u, t_v} P(r_c, u, v, t_u, t_v) r_c}{\sum_{r_c, t_u, t_v} P(r_c, u, v, t_u, t_v)} \tag{4}$$

In other words, this method use the conditional probability $P(r_c|u, v)$ as weight for each possible ratings. It predicts the rating score of each criterion independently. However, the model estimates parameters using the training data where ratings scores on criteria jointly appear, so the correlation among the scores on criteria is incorporated into the estimated parameters.

Table 1 Performance of 3-topics Bayesian model comparison for each evaluation matric, prediction method, and criterion with p-value from paired t-test

Metric-Method	Overall	Story	Acting	Visual	Direction	$p-value$
MAE-B	**2.8581**	**2.7142**	**2.5258**	**2.7777**	2.5190	0.884026
MAE-R	2.8909	2.7349	2.5336	2.7976	**2.5134**	
$P@3$-B	0.75167	0.75750	0.78917	0.74000	0.78833	0.545398
$P@3$-R	**0.76083**	**0.75917**	**0.79250**	**0.75333**	**0.80667**	
$P@5$-B	0.75508	0.75450	0.78958	0.74683	0.80592	0.818035
$P@5$-R	**0.75825**	**0.75917**	**0.79258**	**0.75133**	**0.80992**	
$P@10$-B	**0.75958**	0.75533	0.79008	0.75017	0.80792	0.993486
$P@10$-R	0.75808	**0.75700**	0.79008	**0.75067**	0.80792	
$NDCG_3$-B	0.01131	0.00850	0.00475	0.00445	0.00385	0.205093
$NDCG_3$-R	**0.02212**	**0.00964**	**0.00823**	**0.00747**	**0.00726**	
$NDCG_5$-B	0.00459	0.00537	0.00449	0.00359	0.00355	0.00182558
$NDCG_5$-R	**0.00676**	**0.00623**	**0.00633**	**0.00583**	**0.00556**	
$NDCG_{10}$-B	0.00286	0.00367	0.00293	0.00259	0.00234	0.00188234
$NDCG_{10}$-R	**0.00423**	**0.00420**	**0.00426**	**0.00387**	**0.00364**	

B : Bayesian Estimation. R : Multi-Linear Regression.

Table 2 Performance of 8-topics Bayesian model comparison for each evaluation metric, prediction method, and criterion with p-value from paired t-test

Metric-Method	Overall	Story	Acting	Visual	Direction	$p-value$
MAE-B	**2.8539**	**2.7142**	**2.5255**	2.7777	2.5188	0.988169
MAE-R	2.8762	2.7227	2.5273	**2.7690**	**2.5023**	
$P@3$-B	0.74750	0.75417	0.78917	0.75000	0.80167	0.746195
$P@3$-R	**0.75750**	**0.75750**	**0.79333**	**0.75500**	**0.80500**	
$P@5$-B	0.75750	0.75825	0.79208	0.75308	0.81033	0.891801
$P@5$-R	**0.76425**	**0.76033**	**0.79275**	**0.75333**	**0.81175**	
$P@10$-B	0.75800	0.75508	**0.78958**	0.75042	**0.80933**	0.995202
$P@10$-R	**0.75808**	**0.75617**	0.78925	**0.75067**	0.80875	
$NDCG_3$-B	0.01113	0.00480	0.00518	0.00444	0.00379	0.142862
$NDCG_3$-R	**0.01687**	**0.00831**	**0.00748**	**0.00716**	**0.00797**	
$NDCG_5$-B	0.00462	0.00410	0.00476	0.00394	0.00326	**0.000181415**
$NDCG_5$-R	**0.00629**	**0.00567**	**0.00600**	**0.00600**	**0.00604**	
$NDCG_{10}$-B	0.00289	0.00271	0.00308	0.00252	0.00238	$\mathbf{3.63023 \times 10^{-5}}$
$NDCG_{10}$-R	**0.00381**	**0.00369**	**0.00411**	**0.00398**	**0.00394**	

B : Bayesian Estimation. R : Multi-Linear Regression.

3.2 Multi-Linear Regression

It is the extension of simple linear regression where the predictor variable has more than a variable which can be written as vector $\mathbf{x}$, with scalar response variable y. Mathematically, this technique assume that there is a relationship between predictor and response variable which can be written as:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_n x_{in} \quad (5)$$

where n and p are number of all data and predictor variables respectively. β is n-dimensional vector called a regression coefficient. For implementing this techniques in prediction step of our multi-criteria model, we performed this technique for each criterion separately as well as the Bayesian estimation method in section 3.1. For each i-th training data, we used $P(u_i|t_{u_i})$ and $P(v_i|t_{v_i})$ which corresponding to the user u_i and item v_i in each. For example, in 5-topics Bayesian model, for each user and item in i-th training data, $P(u_i|t_{u_i})$ and $P(v_i|t_{v_i})$ are 5-dimensional vectors. Therefore, the predictor vector $\mathbf{x}_i$ consisted of $P(u_i|t_{u_i})$ and $P(v_i|t_{v_i})$. Then, $\mathbf{x}_i$ became 10-dimensional vector, and y_i is rating score for each criterion.

In order to approximate vector β, there are many way to do so, such as: maximum likelihood estimation, Theil-Sen estimator. Anyway, we just use a simple one, an ordinary least square (OLS) which can be computed as follows:

$$\hat{\beta} = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}^T\mathbf{y} = (\frac{1}{n}\sum \mathbf{x}_i\mathbf{x}_i^T)^{-1}(\frac{1}{n}\sum \mathbf{x}_i y_i) \quad (6)$$

4. Experiments and Results

4.1 Dataset

We use data gathered from Yahoo! Movie website which has an instruction for collecting some users preference on each movie such as: overall ratings, and ratings for another four criteria (story, acting, directing, and visual effects). The users' rating for each criterion are stored in ordered value A+ to F which are the most and the least preferable values. We convert those value into numeric value 13 to 1 respectively. Our dataset consisted of data for 200 users and 1358 movies with 2550 ratings, which were separated into training set (70% of all ratings) and test set (30% of all ratings). We performed experiments using various combination of parameters, in which the numbers of user and item topics were 3, 8, and 16; while the numbers of ranked items in the list were 3, 5, and 10.

4.2 Evaluation Metrices

For evaluating our methods, we use three well-known metrics used in evaluating recommender system. The first one is mean absolute error (MAE) which compute difference between predicted rating and the real. The higher value of MAE denoted there is so many different between predicted ratings and the real which is not good. In contrast, another two metrics evaluate the method using ranked item list for each user by ratings. We considered

Table 3 Performance of 16-topics Bayesian model comparison for each evaluation metric, prediction method, and criterion with p-value from paired t-test

Metric-Method	Overall	Story	Acting	Visual	Direction	$p-value$
MAE-B	**2.8788**	**2.7142**	2.5254	**2.7777**	2.5187	0.926172
MAE-R	2.8805	2.7177	**2.5136**	2.7787	**2.4741**	
$P@3$-B	0.73417	0.75167	0.78500	0.73500	0.79167	0.369232
$P@3$-R	**0.75917**	**0.75750**	**0.79417**	**0.75500**	**0.81167**	
$P@5$-B	0.75525	0.75300	0.78775	**0.74833**	0.80592	0.975287
$P@5$-R	**0.75600**	**0.75350**	**0.78833**	0.74808	**0.80692**	
$P@10$-B	**0.75808**	0.75267	0.78925	**0.75067**	0.80775	0.976743
$P@10$-R	0.75750	**0.75517**	**0.78983**	0.75042	**0.80792**	
$NDCG_3$-B	0.01189	0.00523	0.00659	0.00507	0.00523	0.398517
$NDCG_3$-R	**0.01865**	**0.00754**	**0.00762**	**0.00631**	**0.00598**	
$NDCG_5$-B	0.00488	0.00414	0.00596	0.00421	0.00435	0.065234
$NDCG_5$-R	**0.00648**	**0.00539**	**0.00640**	**0.00504**	**0.00514**	
$NDCG_{10}$-B	0.00301	0.00287	0.00360	0.00285	0.00282	0.010874
$NDCG_{10}$-R	**0.00392**	**0.00363**	**0.00410**	**0.00351**	**0.00332**	

B : Bayesian Estimation. R : Multi-Linear Regression.

user thought in terms of:

- users have positive thought with a movie if they give it a rating score ranging between 7 and 13, and
- users have negative thought with a movie if they give it a rating score ranging between 1 and 6.

The second metric was the precision of the top-k ranked items list:

$$P@k = \frac{L_k}{k} \tag{7}$$

where L_k is the number of true-positive items. In other words, the number of items in the predicted ranked item list which the user also like in real. Note that the value of P@k ranged from 0 to 1, and a higher value referred to a better precision.

The final metric we used was the normal discounted cumulative gain (NDCG). Suppose a recommender system produces a ranked item list $(I_1, I_2, \ldots, I_k)$ for a user u and u's rating for the i-th item (I_i) is r_i. For user u, the DCG for the top-k ranked items list is defined as:

$$DCG_{uk} = \sum_{i=1}^{k} \frac{2^{r_{uk}} - 1}{log(1+i)} \tag{8}$$

Let DCG^*_{uk} denote the highest DCG_{uk} value among the possible ranked item lists. The $NDCG_k$ is then defined as DCG_{uk} divided by DCG^*_{uk}. For a set U of users, we evaluated the performance of a recommender system as the averaged $NDCG_k$ values as follows:

$$NDCG_k = \frac{1}{|U|} \sum_{u \in U} \frac{DCG_{uk}}{DCG^*_{uk}} \tag{9}$$

Note that the value of the NDCG also ranged from 0 to 1, and a method's performance was good as the $NDCG_k$ was high.

4.3 Regression versus Bayesian Estimation

In this paper, we focus on a comparison of predicted method for our MC Bayesian model which are Bayesian estimation and multi linear regression as wrote in section 3.1 and 3.2 respectively. In our recent paper, we introduced this model against Sahoo's model [7], and showed that our model outperformed the other one for both evaluation metric: $P@k$ and $NDCG_k$ for each top-k items list. However, the difference is a bit small, a paired t-test with a significance level set to 0.1 ($\alpha = 0.1$) showed that our proposed method improves the performance significantly. After that, we tried to improve our model to have more difference and significant than the old one. Therefore, we focused on comparison of prediction procedures which are Bayesian estimation and Multi-linear regression.

Table 1, 2, and 3 showed that using multi-linear regression made a model's performance a bit worse at MAE, which makes our objective failed. However, we noticed unexpected advantages. With multi-linear regression, the performance evaluated by $P@k$ is slightly improved with a few item in the list (k). In the meanwhile, $NDCG_k$ value have a high significant ($\alpha = 0.01$) improved when using a large number of items in the list with 8 topics.

5. Conclusion

We have experimented a new prediction procedure for Bayesian model-based recommender system by using multi-linear regression. We expect new method will make the error evaluated by MAE smaller than the old one. According to the experiment result, it does not meet our expectation, but it improves the error evaluate by $NDCG_k$ instead. We still plan to extend our model in various way, and have some experiments for finding the method which is the most suitable for our model.

References

[1] Samatthiyadikun, P., Takasu, A., and Maneeroj, S.: Multicriteria Collaborative Filtering by Bayesian Model-based User Profiling, IRI2012, pp. 124–131 (2012).

[2] Si, L. and Jin, R.: Flexible Mixture Model for Collaborative Filtering *ICML-2003 proceedings*, pp 8 (2003).

[3] Adomavicius, G. and Kwon, Y.: New recommendation techniques for multicriteria rating systems, IEEE Intelligent Systems, Vol.22, issue 3, pp. 48-—55, (2006).

[4] Manouselis, N. and Costopoulou, C.: Analysis and classification of multi-criteria recommender systems, World Wide Web, Vol.10, issue 4, pp. 415-—441, (2007).

[5] Tangphoklang, P., Maneeroj, S., and Takasu, A.: A Novel Weighting Scheme for a Multi-Criteria Rating Recommender System, IADIS, pp. 21-—29, (2011).

[6] Blei, D.M.., Ng, A.Y., and Jordan, M.I.: Latent dirichlet allocation, IADIS, Vo.3, pp. 93-—1022, (2003).

[7] Sahoo, N., Krishnan, R., Duncan, G., and Callan, J.: On multi-component rating and collaborative filtering for recommender systems: The case of yahoo! movies, *Information Systems Research*, (to appear).