Effect of the Number of Users and Bias of Users' Preference on Recommender Systems

Akihiro Yamashita[1], Hidenori Kawamura[1],
Hiroyuki Iizuka[2], and Azuma Ohuchi[1]

[1] Graduate School of Information Science and Technology, Hokkaido University
North 14, West 9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan
{yama,kawamura,ohuchi}@complex.eng.hokudai.ac.jp
http://harmo.complex.eng.hokudai.ac.jp/
[2] Department of Media Architecture, Future University-Hakodate
116-2 Kamedanakano-cho, Hakodate, Hokkaido, 041-8655, Japan
ezca@fun.ac.jp

Abstract. Recommender System provides certain products adapted to a target user, from a large number of products. One of the most successful recommendation algorithms is Collaborative Filtering, and it is used in many websites. However, the recommendation result is influenced by community characteristics such as the number of users and bias of users' preference, because the system uses ratings of products by the users at the recommendation.
In this paper, we evaluate an effect of community characteristics on recommender system, using multi-agent based simulation. The results show that a certain number of ratings are necessary to effective recommendation based on collaborative filtering. Moreover, the results also indicate that the number of necessary ratings for recommendation depends on the number of users and bias of the users' preference.

Key words: Recommender System, Collaborative Filtering, users' preference, Multi-agent Simulation,

1 Introduction

Many E-commerce sites offer hundreds of thousands or even millions of products for sale. Although various products can be obtained on the web, a customer has to invest time and effort to find satisfactory products. A recommender system provides personalized recommendation for each customer to reduce the problem [1, 2]. One of the most successful recommendation technologies is Collaborative Filtering [5, 6, 7], and is used in many websites[3]. A recommender system based on collaborative filtering try to predict a *utility* of products for a target user by using the ratings of products previously received from other users. The utility denotes satisfaction level to a product by the user.

In order to make accurate recommendations, the system has to first learn the preference or interests of the target user from the ratings given by the user. When there are few ratings, the system cannot provide beneficial recommendation. This problem is known as *new user problem* or *cold start problem*[4]. Therefore, each user has to put in a certain number of ratings to the system, especially; this problem is serious in initial operation. One of the simplest solutions of the problem is to recommend the most popular product in all users as a substitute for personalized recommendation. However, it is difficult to estimate that how much ratings should be gathered, because, a recommendation effect is influenced by a characteristics of the community, i.e. the number of users and a bias of users' preference are important to a recommendation methodology.

The purpose of this paper is to evaluate the effect of community characteristics on recommender system, and we evaluate the effect using multi-agent simulation. In this paper, we focus on the number of users and a distribution of users' preference.

This paper is organized as follows: Section2 describes model settings based on multi-agent simulation. Then, Section3 explains procedure of experiments using the simulation model, and Section 4 shows the results and discussion. Finally, Section 5 concludes the paper.

2 Simulation Model

In order to understand the recommendation process, a multiagent-based simulation model was built and applied. In this model, it was assumed that users who have different interest, request recommendation for certain products from the system. Each user has his/her own preference, and each product has its own feature. Generally, if a certain user finds the product suitable for his/her preference, the user will reach high satisfaction level. The degree of such satisfaction can be called *utility*. Thus, the role of a recommender system can be rephrased as recommending certain products for the target user so that the utility will be improved.

Generally, a recommender system receives information from a user about the products the user is interested in. If the system obtains utility directly, it can accurately measure a satisfaction level. However, the only collectible information by the system is *rating* in this model. The rating is coming from the utility, i.e. the utility, which exists in the user's mind, is converted into rating in order to be able to put in the recommender system as feedback information. If the correspondence between utility and rating is optimal, the system will give the most accurate information.

There are many websites collecting ratings about products, using a form interface such as two-choice question (like or dislike) or five-grade rating (from one star to five stars). In this paper, five-grade rating was adopted as the form interface. Details of the model are shown below.

2.1 Model Components

The simulation model has three components: user, product and recommender system. This model can be formulated as follows: Let $U = \{u_i | i = 1, 2, ..., N_u\}$

be the set of all users, let $C = \{c_j | j = 1, 2, ..., N_c\}$ be the set of all products and let RS be a recommender system. The preference of each user u_i is represented by the vector $\mathbf{p_i} = (p_{i1}, p_{i2}, \ldots, p_{iN_f})$, and the feature of each product c_j is represented by the vector $\mathbf{v_j} = (v_{j1}, v_{j2}, \ldots, v_{jN_f})$, where N_f is the dimension of $\mathbf{p_i}$ and $\mathbf{v_j}$.

Intuitively, feature values (elements of the $\mathbf{v_j}$) may be related to shape of product, color of product, etc. Each element of vector $\mathbf{p_i}$ denotes the most ideal product for user u_i. When, it comes to define utility, the distance between the preference vector $\mathbf{p_i}$ and the feature vector $\mathbf{v_j}$ may be calculated. If the distance is small, the utility is high. Thus, the dimension of the two vectors should be equal.

2.2 Utility

Utility function $f_{utility}$ is a function to calculate the utility value s_{ij} from the preference vector $\mathbf{p_i}$ and the feature vector $\mathbf{v_j}$. The function is formulated as follows:

$$s_{ij} = f_{utility}(\mathbf{p_i}, \mathbf{v_j})$$

This function can be designed in several ways; however, there are certain rules that have to be followed: the function has to be monotonically decreasing function, and it has to be inversely proportional to the distance of the two vectors. The most important factor for the user u_i is to find a product with the closest utility to his/her preference. This information can be obtained from calculating the differences between utilities in the area near the preference vector. In this paper, an exponential function is used to accentuate of the utilities, and that can be formulated as follows:

$$\begin{aligned} f_{utility}(\mathbf{p_i}, \mathbf{v_j}) &= exp\big(-\alpha ||\mathbf{p_i} - \mathbf{v_j}||\big) \\ &= exp\big(-\alpha \sqrt{\sum_{k=1}^{N_f} (p_{ik} - v_{jk})^2}\big) \end{aligned}$$

where α is adjusted by the value range of the vectors. α has to be a positive constant, and in this experiment, it was set as $\alpha = 0.5$. The value range of the function is $(0, 1]$.

2.3 Rating

In this model, let R be the set of all product ratings by users, and let rating $r_{ij} \in R$ be the rating for product $c_j \in C$ by user $u_i \in U$. Even if different users have same utilities about certain product, different ratings can be generated. However, it is preferable to decrease this difference as much as possible, for instance by presenting detailed explanation on a website about the rating process. In this model, an ideal setting is assumed, i.e. every user decides his/her rating based on the same rating process f_{rating}.

As described above, five-grade rating was used in this model. Thus, the function f_{rating} has four thresholds. Users should rate uniformly, without biasing to a low or high rating. This histogram on figure1 shows the distribution of utility s_{ij}, when preference vector $\mathbf{p_i}$ and feature vector $\mathbf{v_j}$ are given at a uniform random number.

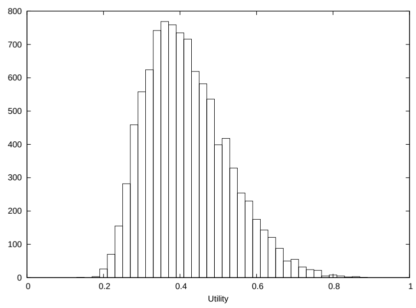

Fig. 1. Utility distribution when preference vector $\mathbf{p_i}$ and feature vector $\mathbf{v_j}$ are given at a uniform random number.

The utilities on this histogram were divided into five equal areas, and the boundary values were utilized as the thresholds of function f_{rating}. Accordingly, function f_{rating} is defined in this model as follows:

$$r_{ij} = f_{rating}(s_{ij}) = \begin{cases} 1 & (s_{ij} \leq 0.33) \\ 2 & (0.33 < s_{ij} \leq 0.38) \\ 3 & (0.38 < s_{ij} \leq 0.44) \\ 4 & (0.44 < s_{ij} \leq 0.51) \\ 5 & (0.51 < s_{ij}) \end{cases}$$

2.4 Recommendation Algorithm

The recommendation problem is reduced to the problem of estimating ratings for the products that have not been seen by a user [4]. In most cases, users rate only a little subset of items, in other words, most of the rating r_{ij} is unapparent. Therefore, the role of RS is to estimate unapparent ratings. The product with the highest estimation value of rating will be offered by recommender systems. In this paper, the following recommendation algorithms were experimented.

Random Recommendation: Random Recommendation algorithm provides the random estimations of the unapparent ratings. In fact, the system recommends a randomly-selected product to the target user. This algorithm is the basis of evaluation of other recommendation algorithms.

Popular Products Recommendation: Popular Products Recommendation recommends the product that receives the highest popularity from all users. The system calculates the mean value of the evaluation of all products, and the product with the highest mean value will be recommended to the target user.

If two or more products have same mean value of the ratings, one of them is chosen at random, and recommended. Mostly, a lot of users are recommended the same product, as the number of ratings increase.

Collaborative Filtering: Collaborative Filtering (CF) estimates the unapparent ratings based on the ratings of other users [3, 6]. When the system estimates ratings for the target user, firstly, CF organizes a set of users who have a history of agreeing with the target user using statistical techniques. In this operation, CF calculates similarities between the target user and the others.

Generally, the similarity between two users is defined by the correlation coefficient of the item rating which either of the users evaluated as follows:

$$sim(u_i, u_j) = \frac{\sum_{a \in C_i \cap C_j} (r_{ia} - \overline{r}_i)(r_{ja} - \overline{r}_j)}{\sqrt{\sum_{a \in C_i \cap C_j} (r_{ia} - \overline{r}_i)^2} \sqrt{\sum_{a \in C_i \cap C_j} (r_{ja} - \overline{r}_j)^2}}$$

$$\text{where} \quad \overline{r}_i = \frac{1}{|C_i|} \sum_{a \in C_i} r_{ia}$$

C_i is the set of products in which user u_i was already rated, and C_{ij} is also the set of products in which both user u_i and user u_j were already rated. If C_{ij} is an empty set, the similarity will be calculated as $sim(u_i, uj) = 0$.

After every similarity between two users has been calculated, an unapparent ratings $\widehat{r}_{i\alpha}$ for a product c_α by user u_i can be described as follows:

$$\widehat{r}_{i\alpha} = \widetilde{r}_i + \frac{\sum_{j \in U_\alpha} sim(u_i, u_j)(r_{j\alpha} - \overline{r}_j)}{\sum_{j \in U_\alpha} |sim(u_i, u_j)|}$$

where U_α is the set of users who rated the product c_α, and $\widetilde{r}_i$ is the average of all ratings that user u_i ever rated as follows:

$$\widetilde{r}_i = \frac{1}{|C_i|} \sum_{a \in C_i} r_{ia}$$

As a result, the system recommends the product c_α with the maximum of rating estimate $\widehat{r}_{i\alpha}$ to user u_i.

In this model, RS recommends a certain product using the algorithm for unapparent ratings $\widehat{r}_{i\alpha}$ based on the correlation coefficient. However, in case of too less ratings, all similarities of user u_i cannot be calculated. Therefore, the rating estimate is defined as $\widehat{r}_{i\alpha} = 3$, and RS makes recommendation at random from among the products which have same estimate.

3 Experiments

3.1 Simulation procedure

Recommender systems based on CF can improve the accuracy of the recommendation as the number of ratings increases. Even if the system can utilize only few ratings, an effective recommendation will be provided as much as possible. When evaluating RS, the relationship between the number of ratings and the accuracy (utilities of all users) should be revealed.

In the following experiments, N_u is a number of users and N_c is a number of products. RS recommends a product each time a rating was received from a user, and then the user rates the recommended product for RS. This interaction is repeated until all of N_u users rate all of N_c products. This operation can be described more in detail as follows:

step 1: Each user picks up one product among all products at random. Then, a utility of the product is calculated, and a rating obtained from the utility is put into RS by each user.

step 2: RS estimates the ratings of the products, which have not been rated yet, using the rating already put in by the users. As described above, three types of recommendation algorithm are utilized for this estimation.

step 3: The product which has the highest rating estimation is recommended to a target user.

step 4: Each user calculates a utility of the product recommended by RS. Since an average of the utilities can be perceived as recommendation accuracy, the average is named $UtilityAverage$. After the rating obtained from the utility is put into RS, the system returns to step2.

3.2 Experimental Settings

Distribution of the Preference Vector: The preference vector $\mathbf{p_i}$ represents the most ideal product of the user. Therefore, the distribution of the vector $\mathbf{p_i}$ intuitively denotes the trend or the vogue among the users. Each element of the vector $\mathbf{p_i}$ is set as random numbers according to the following distribution. Four types of the distribution were applied to investigate an effect of the recommendation by the difference of the distribution. Note that each element of the vector is mutually independent.

- Uniform distribution of range [-1, 1]
- Normal distribution $N_1(\mu = 0, \sigma^2 = 0.5)$
- Normal distribution $N_2(\mu = 0, \sigma^2 = 0.2)$
- Two-peaks distribution (A distribution obtained by randomly selecting either one of the following two normal distributions: $N(\mu = -0.5, \sigma^2 = 0.2)$ or $N(\mu = 0.5, \sigma = 0.2)$)

Uniform distribution represents a situation without trend of the preference, and normal distribution represents a situation where the trend exists. Particularly, the trend is strong when the variance is small in the normal distribution. Furthermore, Tow-peaks distribution represents a situation that the users form several group by their preference. The number of groups is calculated by 2^d where d is the dimension of the vector $\mathbf{p_i}$, because random numbers are independently given to each element.

Constants: Table 1 shows the experiment constants that we used.

Table 1. Experiment Constants

Number of users N_u	100, 500, 1000, 5000, 10000
Number of products N_c	1000
Dimension of vector p_i	5
Dimension of vector v_j	5
α of utility function $f_{utility}$	0.5

4 Results and Discussion

Distribution of the Feature Vector: In this simulation, various products are provided as the recommendation candidates, for investigating an effects of user preference on that recommendation performance. If there is bias in the distribution of feature vectors $\mathbf{v_j}$, it also effects the recommendation result. However, in this experiments, we focus on the effect of the distribution of preference vector. Thus, we eliminate the effect of the distribution of $\mathbf{v_j}$ on the recommendation result by providing the feature vector $\mathbf{v_j}$ at random. More formally, the vector $\mathbf{v_j}$ is set as numbers according to the uniform distribution of range$[-1, 1]$

4.1 Experiments with Each Recommendation Algorithm

The following figures (from figure2 to figure5) show experimental results by each recommendation algorithm, in four types of distribution of the preference vector $\mathbf{p_i}$. The figures are results in case of $N_u = 1000$, and these figures were made by one-shot experiment. However, if the number of users increases, fluctuation at each experiment is few, because the graphs show UtilityAverage. An advantage of the popular products recommendation and the collaborative filtering can be observed by comparing with the random recommendation.

CF is equal to the random recommendation when there are few ratings, and it begin to produce the effectiveness from the vicinity of 30 ratings. The effect of CF is declined in the latter half of the figures, because it previously recommend certain products that matches to the users preference. Popular Product Recommendation produces better UtilityAverage than CF in early step, and a cross

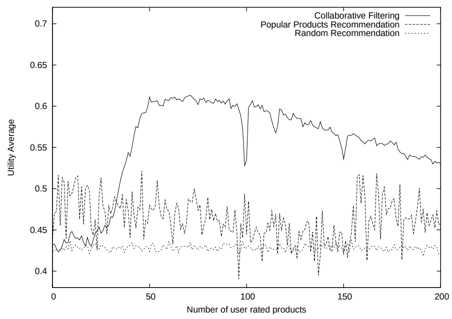

Fig. 2. Uniform distribution

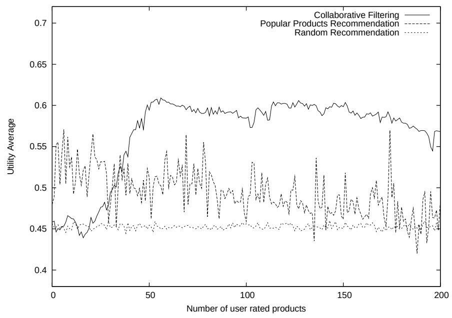

Fig. 3. Normal distribution $N_1(\sigma^2 = 0.5)$

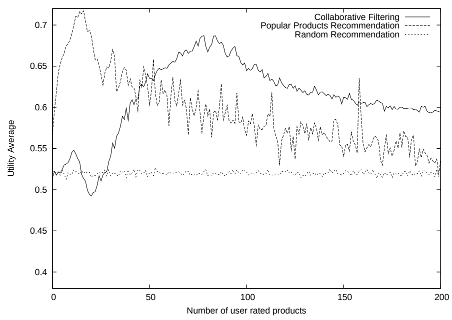

Fig. 4. Normal distribution $N_2(\sigma^2 = 0.2)$

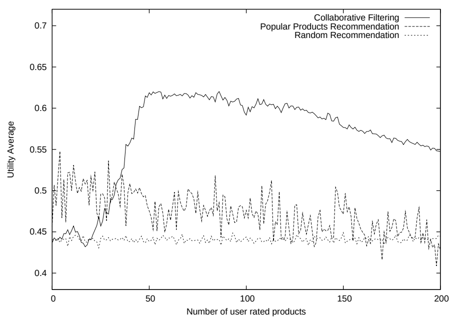

Fig. 5. Two-peaks distribution

point between the line of Popular Product Recommendation and CF is about 30 ratings.

When we compare the difference by the distribution of the preference vector, first of all, UtilityAverage of random recommendation is different in four figures. The figures show that, if the distribution of the preference vector is N_2, UtilityAverage of random recommendation is most high, because UtilityAverage rises when the vector is gathered in the center. Moreover, in the N_2 distribution, Popular Products Recommendation produce high UtilityAverage, because entire popularity was gained by several products, which have a feature vector toward near the center. However, CF is the highest result in the latter half of any figures.

4.2 Relationship between number of users and recommendation accuracy

The following figures (from figure6 to figure9) shows the experimental results of CF based recommendation in various numbers of users. Each figure illustrates the result in the situation of each distribution of preference vector $\mathbf{p_i}$. The figures were also made by one-shot experiment. It is necessary to repeat similar experiments to obtain a more accurate experimental result. However, the tendency to the recommendation accuracy in an increase in the number of users can be obtained from these results.

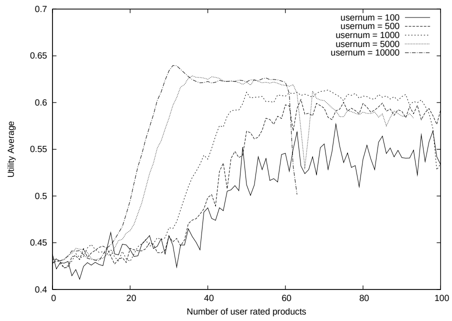

Fig. 6. Uniform distribution

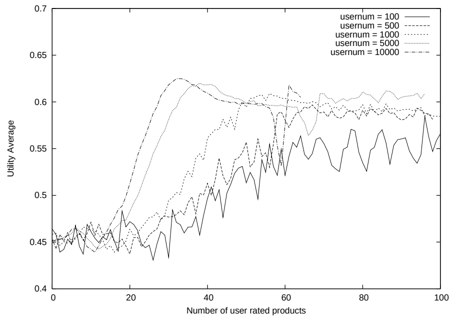

Fig. 7. Normal distribution $N_1(\sigma^2 = 0.5)$

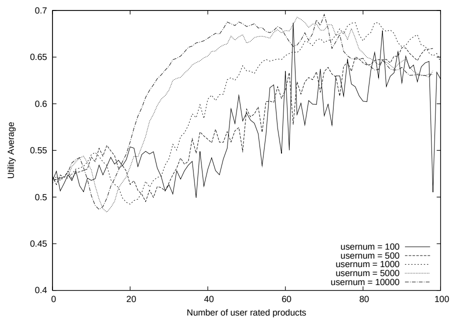

Fig. 8. Normal distribution $N_2(\sigma^2 = 0.2)$

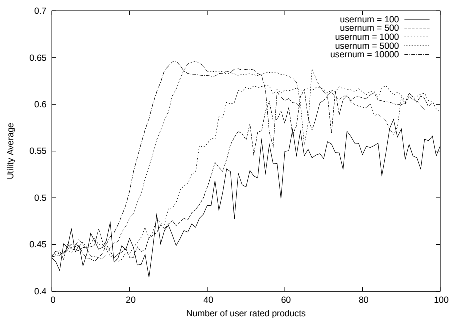

Fig. 9. Two-peaks distribution

As discussed previously, CF in early step is equal to a random recommendation. We should compare the recommendation effectiveness by difference of UtilityAverage between the value at 0 rating and the maximum value.

Each figure indicates that if there are a lot of users, UtilityAverage will be high; moreover, an effective recommendation in a less ratings is possible. For instance, if 10,000 users use the system and each user inputs 20 ratings, CF provides certain effective recommendation. However, even if the system has a lot of users, CF cannot recommend preferable products, when each user rates only less than about 20 products. In this case, the system should provide popular product recommendation together.

In addition, when the number of users is little, the UtilityAverage is not high, even if many ratings are given by each user. In a word, it is important in the improvement of recommendation accuracy to gather many users.

5 Conclusion

In this paper, we built a multi-agent based simulation model for investigation of the effectiveness of a recommendation system. Especially, we investigated collaborative filtering which is used in many websites today.

Recommender system recommends certain products based on a lot of product's ratings which were previously rated by the users. Thus, the accuracy of CF based recommendation depends on a community characteristics, such as the number of users or a distribution of users' preference. In this research, we examined the effect of community characteristics on the recommender system. And we investigated the difference between popular product recommendation and CF.

The results show that, popular product recommendation provides higher utility than CF, if there are few ratings. And the CF overtakes if a lot of ratings are utilized. Moreover, it was confirmed that the recommendation accuracy is improved when there are a lot of users.

References

1. P. Resnick, and H. Varian,: Recommender Systems, Comm. of the ACM, Vol.40, No.3, pp.56-89 (1997)
2. E. Reategui, J. A. Campbell, R. Torres "Using Item Descriptors in Recommender Systems" AAAI Workshop on Semantic Web Personalization, San Jose, USA, 2004.
3. G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering" IEEE Internet Computing, Jan. / Feb. 2003.
4. G. Adomavicius and A. Tuzhilin "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions" IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.6, JUNE 2005 pp.734-749.
5. B. Sarwar, G. Karypis, Joseph. Konstan and J. Riedl, "Analysis of Recommendation Algorithms for E-Commerce", Proceedings of the 2nd ACM conference on Electronic commerce, p.158-167, October 17-20, 2000, Minneapolis, Minnesota, United States
6. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl "GroupLens: An Open Architecture for Collaborative Filtering of Netnews" In CSCW '94: Conference on Computer Supported Cooperative Work (Chapel Hill, 1994), ACM, pp. 175-186.
7. N.Good, J. B. Konstan, J. A. Borchers, B. Sarwar, J. Herlocker and J. Riedl "Combining Collaborative Filtering with Personal Agents for Better Recommendations" Proceedings of the 1999 Conference of the American Association for Artificial Intelligence (AAAI-99), pp. 439-446. (1999)
8. Hung-Wen Tung and Von-Wun Soo "A Personalized Restaurant Recommender Agent for Mobile E-Service", Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)