Journal of The Korea Society of Computer and Information
Vol. 23 No. 12, pp. 219-226, December 2018

www.ksci.re.kr
https://doi.org/10.9708/jksci.2018.23.12.219

Development of a Personalized Similarity Measure using Genetic Algorithms for Collaborative Filtering

Soojung Lee*

Abstract

Collaborative filtering has been most popular approach to recommend items in online recommender systems. However, collaborative filtering is known to suffer from data sparsity problem. As a simple way to overcome this problem in literature, Jaccard index has been adopted to combine with the existing similarity measures. We analyze performance of such combination in various data environments. We also find optimal weights of factors in the combination using a genetic algorithm to formulate a similarity measure. Furthermore, optimal weights are searched for each user independently, in order to reflect each user's different rating behavior. Performance of the resulting personalized similarity measure is examined using two datasets with different data characteristics. It presents overall superiority to previous measures in terms of recommendation and prediction qualities regardless of the characteristics of the data environment.

▸Keyword: Collaborative Filtering, Recommender System, Similarity Measure, Genetic Algorithm, Data Sparsity Problem

I. Introduction

As Internet users of these days are overloaded with information which are mostly irrelevant to what users seek for, recommender systems have emerged and been helpful in finding information. Various types of recommender systems have been developed so far, where the fundamental ones might be content-based and collaborative filtering systems [1][2].

Content-based filtering (CBF) maintains a user profile so that recommended items have content similar to the profile. One of the main drawbacks of CBF is the serendipity problem, since items with content different from the profile cannot be recommended. Collaborative filtering (CF) can overcome this problem, as it references other like-minded users to recommend items. It is further categorized into two methods, user-based and item-based. User-based CF recommends items to a user based on the rating history of other users having similar preferences to the user's. Item-based CF recommends items similar to items for which the user has given preferences.

There are other variants of recommender systems, such as hybrid filtering, demographic filtering, and social filtering[1]. Hybrid filtering basically combines CBF and CF using various hybridization methods. Demographic filtering utilizes the user's demographic attributes to produce recommendations, under the assumption that users with common attributes will also have common preferences. Recently, social filtering has been increasingly of interest with the development of web 2.0. It basically exploits trust/distrust concepts from identifying social networks and aims to improve recommendations made by traditional filtering [3].

Although several techniques for recommender systems

• First Author: Soojung Lee, Corresponding Author: Soojung Lee
*Soojung Lee (sjlee@ginue.ac.kr), Dept. of Computer Education, Gyeongin National University of Education

• Received: 2018. 10. 15, Revised: 2018. 11. 05, Accepted: 2018. 11. 08.

have been devised, CF systems are most well-known and successfully implemented in many commercial systems, most popular of which may be Amazon.com[2]. However, CF suffers from data sparsity problem which causes difficulty in determining like-minded users. This problem may result in unreliable recommendations, as user-based CF calculates similarity between users based on their ratings records. A lot of efforts have been devoted to overcome this problem. We make an in-depth study of a simple and popular approach that uses Jaccard index[4] for data sparsity problem. We suggest a new similarity measure to deal with the limitations of using Jaccard index. Our method aims to discover an optimal similarity measure by using a genetic algorithm. We further apply this suggested approach to each independent user, in order to obtain an optimal similarity measure best fit to each user. Extensive experiments are conducted to investigate performance of the proposed approaches which are compared with that of several previous similarity measures using Jaccard index as well as with that of the traditional ones.

The remainder of this paper is organized as follows: The next section describes existing similarity measures along with technologies related to our study. Section 3 describes the motivation of this work and details of the proposed approach. In Section 4, experimental results are provided. Section 5 concludes this paper.

II. Background

1. Existing Similarity Measures

Performance of CF systems is critically dependent on their underlying similarity measures. In user-based CF, the neighbor users of higher similarity with the current user are taken with higher importance in referring to their ratings. Among many similarity measures developed so far, most representative ones are Pearson correlation, cosine similarity, and the mean squared differences. These also became foundations for later-developed measures such as constrained Pearson correlation, Spearman rank correlation, Kendall's τ correlation and adjusted cosine similarity [1][2].

As the traditional metrics have been thoroughly analyzed to investigate their behaviors[5][6], several metrics to complement their disadvantages are developed with new perspectives. Gao et al. notice that all users have the same weight when computing item relationships in item-based CF[7]. They incorporate the weight of a user, which is calculated using a PageRank-based user ranking approach, into the computation of item similarities. Bobadilla et al. define three types of significances in terms of users, items, and user ratings. For similarity computation, they replace the user ratings with the significances to reflect different importances of ratings[8].

Despite the efforts made to assign a different weight to each rating, inherent problems of CF, i.e., data sparsity and cold-start, still remain. This problem is critical as data sparsity often leads to unreliable similarity values. To handle these problems, Liu et al. proposed a similarity model to improve recommendation performance in cold user conditions[6]. They exploited the context information and integrated several previous metrics together to take into account the global preference of user behavior as well as the local context information of user ratings. Ren et al. introduced an overlap factor to be combined with Pearson correlation and the cosine similarity[9]. These new similarity measures are designed to mitigate the sparsity problem and proved to outperform the corresponding traditional measures through various experiments.

2. Jaccard Index

There have been a lot of efforts to reflect the number of items co-rated by two users on similarity calculation between them. Herlocker et al. weight similarities by the number of common ratings between users/items[10]. Jamali and Ester incorporated a sigmoid function with the input of the number of common users into Pearson correlation[11].

A simpler strategy to overcome data sparsity is to combine Jaccard index[4] with existing similarity measures. In user-based CF, this index reflects the number of items co-rated by two users and is formally defined as

$$Jaccard(u,v)=\frac{|I_u \cap I_v|}{|I_u|+|I_v|-|I_u \cap I_v|}$$

where Iu represents the set of items rated by user u. There have been several studies utilizing Jaccard index for computing similarity. Bobadilla et al. also addressed the data sparsity problem and suggested a similarity metric that incorporates Jaccard index into the mean squared differences[5]. Liu et al. heuristically combined several metrics into a new similarity measure which includes Jaccard index[6].

3. Exploiting Genetic Algorithms

Genetic algorithms (GAs) belong to the class of evolutionary algorithms and mimic the process of natural evolution[12]. In GAs, a set of solutions is evolved into an optimized solution through the principles of natural genetics. These algorithms are well-known to be a most powerful technique for optimization problems including feature selection and weighting tasks.

In literature, GAs have been useful in designing hybrid CF approaches. Hwang et al. proposed a hybrid recommender system where a GA is used to learn personal preferences of customers[13]. Another study working on a movie dataset is published by [14]. It uses information for the user profile such as user demographic characteristics as well as genre preferences and ratings. Each feature in the profile is weighted through a GA, which is then used to calculate similarity between users. Sadhasivam et al. proposed a recommendation framework that operates on news[15]. It defines the chromosome as consisting of user profile, group interest, location awareness, and location sentiments, whose weights are identified by a GA for personalized news recommendation.

In the work of [16], the chromosome consists of user/item similarity and the threshold of user/item similarity, each of whose weights is set by running a GA. Obtaining an optimal similarity function is the purpose of the study proposed by Bobadilla et al.[17]. They make use of rating differences between users, whose weights are optimized by a GA. The final similarity metric is defined as a function of these differences along with the corresponding weights.

While GAs have mostly been employed to find optimized weights of user/item attributes, they are used in totally different ways in some of the latest studies[18][19]). Gao and Li proposed a hybrid model where outputs of different recommender systems are integrated by a GA[18]. Meanwhile, Kim and Ahn focused on the user-item matrix to try to condense it to search for only relevant users and items[19], with the intention to ameliorate the scalability problem as well as the sparsity problem. A GA is used to optimize item and user selection simultaneously. This method is experimented with movie datasets and is found to enhance performance over both a conventional CF and other GA-optimized CF methods.

III. The Proposed Scheme

1. Motivation

As described previously, Jaccard index has been utilized as a useful tool to overcome the deficiencies of conventional similarity measures. However, there remains several points to consider as follows.

1. In what kind of data environment, would the combination of Jaccard index be most effective? That is, is the performance improvement distinguishable by the data characteristics such as data sparsity and rating scales?

2. Is the degree of performance improvement different according to the similarity measure combined with Jaccard index?

3. How much importance should be imposed on each factor in the combination for best performance?

To the best of our knowledge, there has been little research on the above questions in literature. Bobadilla et al. suggested a similarity measure that combines the mean squared differences (MSD) with Jaccard index[5]. Their experiment results show that their measure outperforms Pearson correlation in small scaled datasets, but not in a larger scaled set of FilmAffinity. Also, their work does not provide an answer to the third question above, since it suggests the multiplication of MSD and Jaccard index with equal weights. In [6], Liu et al. proposed a metric that combines various factors including the variation of Jaccard index. Their metric is experimentally compared with many other similarity metrics, but the effectiveness of Jaccard index is not analyzed separately, not to mention its optimal weight in the combination.

We aim to answer all three questions above. Furthermore, we search for optimal weights of factors independently for each user as well as for all the users using a GA, since users should have different optimal similarity functions as their rating behaviors are different.

2. System Overview

The system operates in two phases, online and offline. In the online phase the system is to predict ratings of items new to a user and recommends items with the highest predicted ratings. Since rating prediction is made from the nearest neighbors, selection of them critically determines prediction accuracy. Let user v be a nearest neighbor of user u. Then the rating of an item i unrated by user u, $r'_{u,i}$, is predicted using a weighted average as follows[2], where $\overline{r_u}$ is the mean rating of user u:

$$r'_{u,i} = \overline{r_u} + \frac{\sum_v sim(u,v) \times (r_{v,i} - \overline{r_v})}{\sum_v |sim(u,v)|}$$

The goal of our study is to select the nearest neighbors from which the greatest prediction accuracy would be achieved. For this, an optimal similarity measure is obtained using a GA, which is done in the offline phase in consideration of the execution time for the GA. Specifically, our similarity measure, denoted by $\mathrm{SIM_{WGA}}$, between users u and v is defined as

$$SIM_{WGA} = SIM_T(u,v)^{W_S} \bullet Jaccard(u,v)^{W_J}$$

where $\mathrm{SIM_T}$(u,v) is similarity from a traditional similarity measure and w_s and w_J are weights determined by the GA where they sum up to one. This measure is commonly applied to all the users to find their neighbors.

Figure 1 shows the flow of obtaining the optimal w_s and w_J. The whole process is repeated until the stopping criteria are satisfied. If the criteria are not met, the weights are updated by the GA operators and are applied onto the similarity measure to find new nearest neighbors. The stopping criteria are based on how close these predictions are to the real ratings of the users. The GA operations are detailed in the following section.

Additional to the optimal weights commonly applied to all users as above, we search for optimal weights independently for each user, also by using a GA. Thus the similarity measure personalized to each user u, denoted by $\mathrm{SIM_{PGA}}$, is defined analogously to $\mathrm{SIM_{WGA}}$ as

$$SIM_{PGA} = SIM_T(u,v)^{W_{S,u}} \bullet Jaccard(u,v)^{W_{J,u}}$$

between users u and v, where the only difference is the personalized weights $w_{s,u}$ and $w_{J,u}$ for user u.

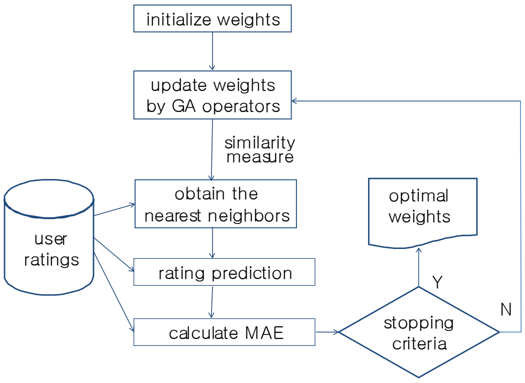


Fig. 1. Workflow for obtaining optimal weights of the proposed similarity measure

3. Genetic Algorithm

A GA usually proceeds as follows. A population of solutions is maintained throughout the execution, where each solution represents a chromosome. Solutions are evolved from generation to generation until a suitable solution satisfying a given criteria named fitness is created.

In our GA, the set of two weights, ws and wJ, comprises an individual of the population. The weight is represented by a bit string $b_i b_{i-1} \ldots b_1 b_0$ that is evaluated as a real number in the range [0, 1]. Implementing it with Nb number of bits, we use the following expression to convert it to a real value[17][20]:

$$w = \frac{\sum_{j=0}^{N_b - 1} 2^j b_j}{2^{N_b} - 1}$$

For each individual of the population, the sum of its weights equals one.

Selection of parents from the population is made based on a selection probability that depends on the fitness function. We chose MAE as the fitness, since it is popularly used for measuring prediction quality in related researches. Let f_i be the fitness of solution i and f_{best} and f_{worst} be the best and worst fitnesses at the current generation, respectively. Then we set the selection probability of solution i as follows:

$$Prob_{S,i} = \frac{1}{e^{c(1-X_i)/T_n}}, \quad \text{c: constant}$$

$$T_n = \ln(1+n), \quad \text{n: number of generations}$$

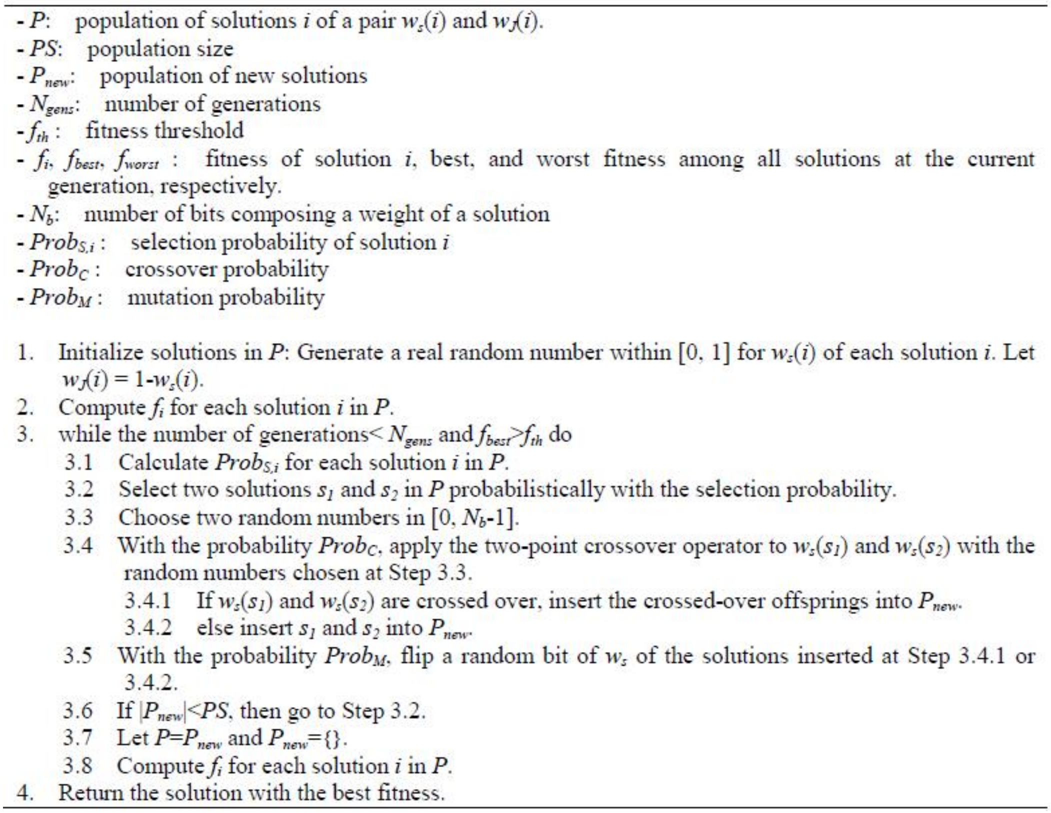
$$X_i = 1 - \frac{f_i - f_{best}}{f_{worst} - f_{best} + \alpha}, \quad \alpha \cong 0$$

In the above definition, we adopted simulated annealing algorithm as a way to prevent the pre-mature problem of the GA[18]. This algorithm allows the solutions with better fitness to be selected with higher probability as time passes. Figure 2 describes the detailed steps of our GA. As the chromosome is composed of two weights w_s and w_J, crossover and mutation operators are applied to w_s's only, and then w_J is calculated as 1.0-w_s.

IV. Performance Experiments

1. Design of Experiments

We performed experiments using MovieLens and Jester



```
- $P$:    population of solutions $i$ of a pair $w_s(i)$ and $w_J(i)$.
- $PS$:    population size
- $P_{new}$:    population of new solutions
- $N_{gens}$:    number of generations
- $f_{th}$ :    fitness threshold
- $f_i$, $f_{best}$, $f_{worst}$ :   fitness of solution $i$, best, and worst fitness among all solutions at the current
    generation, respectively.
- $N_b$:    number of bits composing a weight of a solution
- $Prob_{S,i}$ :    selection probability of solution $i$
- $Prob_C$ :    crossover probability
- $Prob_M$ :    mutation probability

1.  Initialize solutions in $P$: Generate a real random number within [0, 1] for $w_s(i)$ of each solution $i$. Let
    $w_J(i) = 1-w_s(i)$.
2.  Compute $f_i$ for each solution $i$ in $P$.
3.  while the number of generations< $N_{gens}$ and $f_{best}$>$f_{th}$ do
    3.1  Calculate $Prob_{S,i}$ for each solution $i$ in $P$.
    3.2  Select two solutions $s_1$ and $s_2$ in $P$ probabilistically with the selection probability.
    3.3  Choose two random numbers in [0, $N_b$-1].
    3.4  With the probability $Prob_C$, apply the two-point crossover operator to $w_s(s_1)$ and $w_s(s_2)$ with the
         random numbers chosen at Step 3.3.
         3.4.1  If $w_s(s_1)$ and $w_s(s_2)$ are crossed over, insert the crossed-over offsprings into $P_{new}$.
         3.4.2  else insert $s_1$ and $s_2$ into $P_{new}$.
    3.5  With the probability $Prob_M$, flip a random bit of $w_s$ of the solutions inserted at Step 3.4.1 or
         3.4.2.
    3.6  If $|P_{new}|$<$PS$, then go to Step 3.2.
    3.7  Let $P$=$P_{new}$ and $P_{new}$={}.
    3.8  Compute $f_i$ for each solution $i$ in $P$.
4.  Return the solution with the best fitness.
```

Fig. 2. Description of our genetic algorithm

datasets. For measuring prediction quality, we employed MAE, a representative metric for measuring prediction quality[2]. It is defined as $\sum|r_i-r_i'|/N$ for N predictions r_i' for the corresponding real rating r_i of item i. In addition, we evaluated recommendation quality through precision and recall metrics. As a comprehensive indicator, F1, a harmonic mean of precision and recall, will be employed in our presentation of results.

To obtain more reliable results, we conducted 5-fold cross validation, where 80% of ratings data is used for training and the rest for testing. The baseline similarity measures of our experiments are Pearson correlation (COR), the cosine similarity (COS), and the mean squared differences (MSD). Each of these multiplied with Jaccard index, denoted by COR*J, COS*J, and MSD*J, is also evaluated. The proposed SIM_{WGA} is experimented with using each baseline, which is denoted by COR_{WGA}, COS_{WGA}, and MSD_{WGA}. Likewise, COR_{PGA}, COS_{PGA}, and MSD_{PGA} are obtained from SIM_{PGA}.

As for the genetic operation, we set the parameters as in Table 1. For reference, previous works addressing optimal values for crossover and mutation probabilities suggest 0.35~0.85 and 0.01~0.4, respectively[21]. We believe that our purpose of study is not to discover the probabilities producing the optimal performance, but to answer our research questions raised previously. Moreover, from preliminary experiments with various probabilities we could notice a little difference among their results. Hence, we chose one of the values within the range suggested by [21].

Table 1. Parameters for the genetic operation

N_b	P_S	N_{gens}	f_{th} (MovieLens/Jester)	$Prob_C$	$Prob_M$
12	72	20	0.65/2.5	0.45	0.1

The initial population of weights is randomly generated within the range of [0, 1]. Each weight is implemented using 12 bits, in order to have higher precision than 10 bits used in [17]. For the population size, we have referred to the criterion of doubling the number of bits used to represent each individual[22], which is then 48 individuals. But, considering that we have only two weights comprising the individual, we heuristically increased population size to 72 for variability in generating solutions. We used MAE as the fitness function and gave very low thresholds.

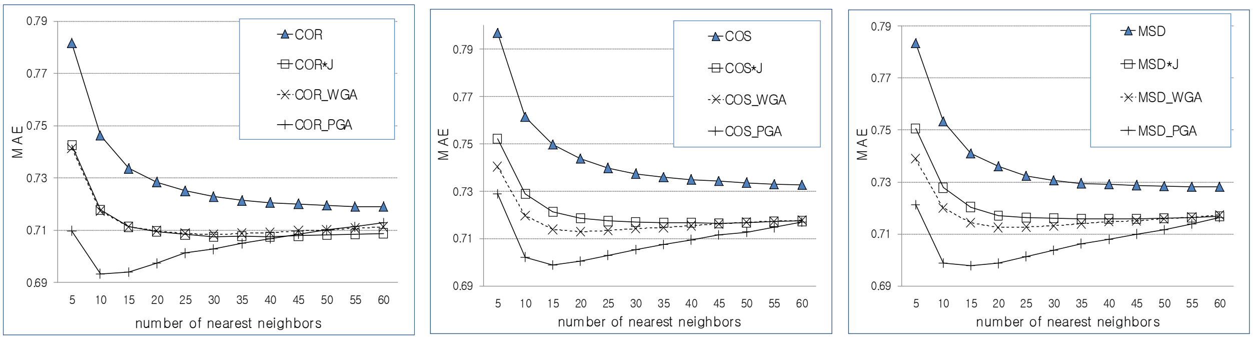


Fig. 3. Comparison of MAE with MovieLens dataset

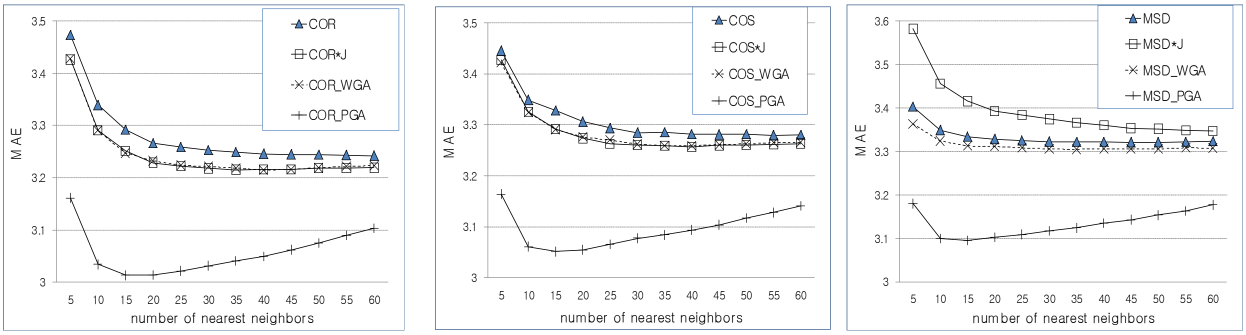


Fig. 4. Comparison of MAE with Jester dataset

2. Results of Prediction Accuracy

Figures 3 and 4 depict MAE of the measures with the two datasets. It is observed that the results of the three conventional measures look very similar with each other with MovieLens as in Figure 3. Also, the results of *J, WGA, and PGA almost look alike across the associated conventional measure type. Note that the conventional measure shows the worst performance, WGA and *J yield noticeable improvement, and that PGA significantly outperforms all the others, regardless of the measure type. Taking a closer look at the COS and MSD results, WGA achieved a bit better performance than *J, while WGA has no advantage over *J in case of COR.

One thing to notice in Figure 3 is a somewhat different behavior of all the three types of PGA. Specifically, PGA tends to have a decreased MAE for NNs less than 15 approximately, and then steadily increased MAE for larger NNs. This behavior is in contrast to the others where their MAEs are getting stabilized with increasing NNs. The reason may be that it is harder to obtain optimal weights suitable for larger NNs than for fewer NNs of a user in PGA strategy. However, the performance of PGA seems to converge to that of WGA in all the three measure types. This is obvious because of the same reason described above; recall that WGA is to find optimal weights common for all users while PGA considers each single user independently. In conclusion, the best results of PGA are better than those of the conventional measures by approximately 3.6~4.6%.

Figure 4 pictures MAE with Jester. Overall, PGA shows consistently superior performance. For the same reason described above for MovieLens, all three types of PGA are observed to yield increased MAEs with NNs. Nevertheless, the best PGA produces about 6.8~7.1% lower MAE than corresponding conventional measure. On the contrary, WGA seems to bring almost no advantage over each conventional measure not to mention *J in case of COR and COS related experiments. This implies that GA execution for finding optimal weights common for all users is not effective in a dense and large-scaled set like Jester, as sufficiently enough data are available for traditional measures to provide reliable neighbors. On the contrary, PGA allows for significant MAE improvement over WGA in all associated measure types, i.e., 6.3~6.4%.

3. Results of Recommendation Accuracy

We presented the F1 results in Figures 5 and 6. We used 80% of the top rating as relevance thresholds: that is, 4 with MovieLens and 6.0 with Jester. The three figures in Figure 5 demonstrate similar behavior with each other. That is, all the measures except the conventional measures are very competitive and

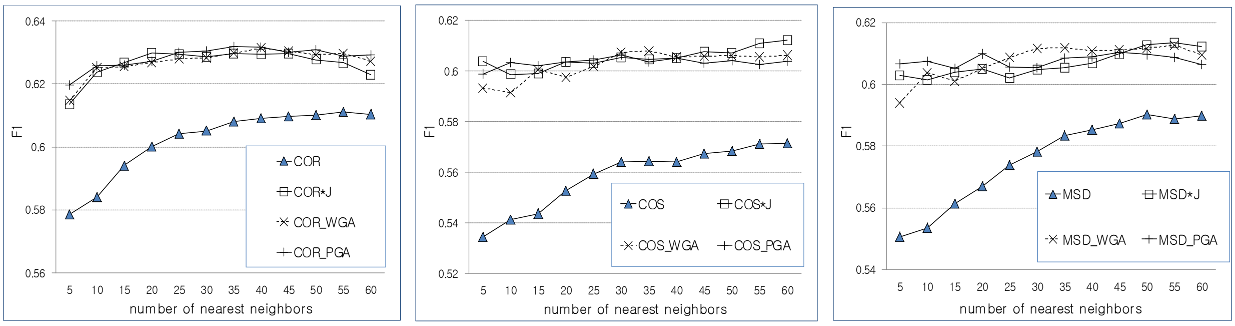


Fig. 5. Comparison of F1 with MovieLens dataset

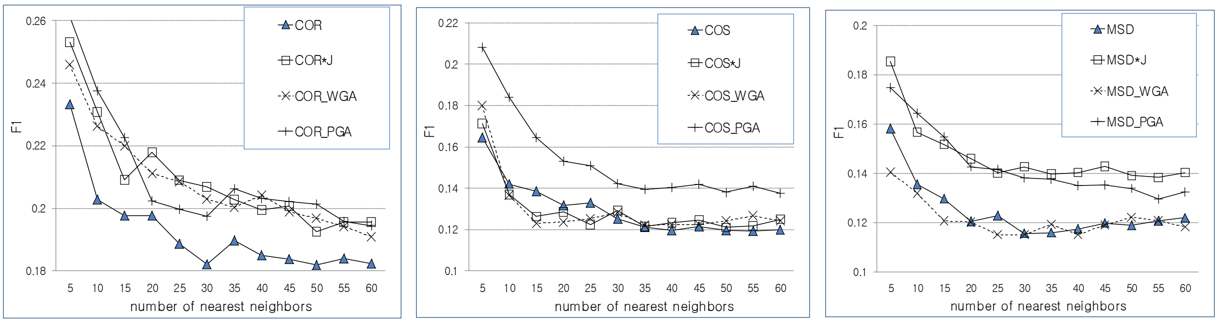


Fig. 6. Comparison of F1 with Jester dataset

significantly defeat the corresponding conventional ones. In Figure 5, it is especially noticeable that PGA performs very comparably to *J or WGA in all the figures, although its MAE is definitely best as shown in Figure 3. It is found that COR is worse by approximately 3.4%, COS by 6.14%, and MSD by 3.4%, than the best F1 result of each corresponding PGA, respectively.

Figure 6 presents F1 performance with Jester. There seems no single winner of all in the three measure types. However, two findings are observed in the results. One is that the conventional measure is generally outperformed by the other measures, especially in COR-related experiments. Another finding is that PGA demonstrates almost the best results throughout three measure types, where $\mathrm{COS}_{\mathrm{PGA}}$ shows distinct improvement over all other corresponding measures. On the contrary, *J and WGA lead to relatively fluctuating performance. To conclude, PGA results in rather lower superiority in F1, compared to the MAE, in all experiments. One reason is that PGA uses MAE as a fitness function instead of F1 in its GA.

V. Conclusions

Combining a traditional similarity measure with Jaccard index has been studied as a useful approach to compute similarity in collaborative filtering systems. We searched for optimal weights in the combined similarity measure by using a genetic algorithm. We further improved our method to find optimal weights for each user to obtain an optimized similarity measure best suited to each.

We conducted extensive experiments and found that the personally optimized similarity measure yielded far better prediction accuracy with all datasets used. With the developed measure implemented, the system will surely provide the user with more satisfiable recommendation list of items. One concern is the computational overload to run a genetic algorithm. A possible solution for this is the periodic offline algorithm execution or applying the proposed measure only to infrequent users. Alternatively, we plan to implement our measure for item-based CF systems where the load for item similarity calculation is typically less than that for user similarity calculation.

REFERENCES

[1] Aamir, M. & Bhusry, M. (2015). Recommendation system: State of the art approach. International Journal of Computer Applications, 120(12), 25-32.

[2] Su, X. & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in Artificial

Intelligence 2009.

[3] Carrer-Neto, W., Hernandez-Alcaraz, M. L., Valencia-Garcia, R., & Garcia-Sanchez, F. (2012). Social knowledge-based recommender system: Application to the movies domain. Expert Systems with Applications, 39(12), 10990-11000.

[4] Lee, S., (2016). Improving performance of Jaccard coefficient for collaborative filtering. Journal of the Korea Society of Computer and Information, 21(11), 121-126.

[5] Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems, 23(6), 520-528.

[6] Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems, 56, 156-166.

[7] Gao, M., Wu, Z., & Jiang, F. (2011). Userrank for item-based collaborative filtering recommendation. Information Processing Letters, 111(9), 440-446.

[8] Bobadilla, J., Hernando, A., Ortega, F., & Gutierrez, A. (2012). Collaborative filtering based on significances. Information Sciences, 185(1), 1-17.

[9] Ren, L., Gu, J. & Xia, W. (2011). A weighted similarity-boosted collaborative filtering approach. Energy Procedia, 13, 9060-9067.

[10] Herlocker, J. L., Konstan, J. A., & Riedl, J. T. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval, 5(4), 287-310.

[11] Jamali, M. & Ester, M. (2009). Trustwalker: A random walk model for combining trust-based and item-based recommendation. Proc. 15th SIGKDD International Conf. Knowledge Discovery and Data Mining (pp. 397-406).

[12] Salehi, M., Kamalabadi, I., & Ghaznavi-Ghoushchi, M. (2013). Attribute-based collaborative filtering using genetic algorithm and weighted c-means algorithm. International Journal of business information systems, 13(3), 265-283.

[13] Hwang, C. S., Su, Y. C., & Tseng, K. C. (2010). Using genetic algorithms for personalized recommendation. Proc. 2nd Int'l Conf. Computational Collective Intelligence: Technologies and Applications (pp. 104-112). Springer-Verlag.

[14] Velez-Langs, O. & Antonio, A. D. (2014). Learning user's characteristics in collaborative filtering through genetic algorithms: Some new results. Soft computing, 309-326.

[15] Sadhasivam, G. S., Saranya, K. G., & Praveen, E. M. (2014). Personalisation of news recommendation using genetic algorithm. Proc. 3rd Int'l Conf. Eco-friendly Computing and Communication Systems (pp. 23-28).

[16] Jia, Y., Ding, Q., Liu, D., Zhang, J., & Zhang, Y. (2014). Collaborative filtering recommendation technology based on genetic algorithm. Applied mechanics and materials, 599/601(2), 1446-1452.

[17] Bobadilla, J., Ortega, F., Hernando, A., & Alcal, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. Knowledge-Based Systems, 24(8), 1310-1316.

[18] Gao, L. Q. & Li, C. (2008). Hybrid personalized recommended model based on genetic algorithm. Int'l Conf. Wireless Communication, Networks and Mobile Computing (pp. 9215-9218).

[19] Kim, K. J. & Ahn, H. (2011). Collaborative filtering with a user-item matrix reduction technique. International Journal of Electronic Commerce, 16(1), 107-128.

[20] Shivhare, H., Gupta, A., & Sharma, S. (2015). Recommender system using fuzzy c-means clustering and genetic algorithm based weighted similarity measure. Int'l Conf. Computer, Communication and Control (pp. 1-8).

[21] San Jose-Revuelta, L. M. (2007). A new adaptive genetic algorithm for fixed channel assignment. Information Sciences, 177(13), 2655-2678.

[22] Alander, J. (1992). On optimal population size of genetic algorithms. Proc. the Computer Systems and Software Engineering Conf. (pp. 65-70). IEEE.

Authors

Soojung Lee received the B.S. degree in Mathematics Education from Ewha University, Korea in 1985. She received M.S. and Ph.D. degrees in Computer Science from Texas A&M University, U.S.A, in 1990 and 1994, respectively Dr. Lee joined the faculty of the Department of Computer Education at Gyeongin National University, Korea, in 1998. She is interested in recommender systems, information filtering, data mining techniques, and computer education.