

Improved Memory-based Collaborative Filtering Using Entropy-based Similarity Measures

Hyeong-Joon Kwon, Tae-Hoon Lee, and Kwang-Seok Hong

School of Information and Communication Engineering, Sungkyunkwan University, Suwon 440-746, South Korea

Email: {katsyuki, ithysk}@skku.edu and kshong@yurim.skku.ac.kr

Abstract— Accuracy of predicting the user preference score is the most important element of collaborative filtering. This paper proposes novel similarity measures using difference score entropy of common rating items between two users. The proposed similarity measures can apply various weights according to the score difference, to evaluate the similarity. We implemented a recommender system using the proposed similarity measures and, experimented on performance with memory-based collaborative filtering. Based on the experimental results, the proposed similarity measures significantly improve the prediction accuracy with respect to existing similarity measures, and we confirmed that the proposed measure is robust to sparse data sets.

Index Terms—similarity, collaborative filtering, entropy

I. INTRODUCTION

The amount of information on the Web is increasing according to the growth of information and communication infrastructure. As a result, recommender systems (RSs) for personalization are required. An RS provides contents or items considering the tastes of individual users. Among the various RSs, collaborative filtering (CF) is the process of filtering for information or patterns using collaborative techniques involving multiple users [1].

The CF predicts the preference score of a user for items he/she had not previously evaluated. To predict the preference score, CF systems use a user-item data set which is comprised of numerical preference scores with a fixed range. Data sets for algorithmic experiments have been provided by many research institutions, including the MovieLens data set (GroupLens project team, Minnesota University), the Jester data set (Berkeley Laboratory for Automation Engineering) and EachMovie (HP/Compaq DEC research center)[2], [3], [4]. Each data set includes its own attributes in their separate forms. For example, the range of reference scores and the genres of each movie. Many researchers and laboratories focus on improving prediction accuracy with such data sets. According to a survey paper of CF systems, CF can be divided into two types [1]. They are memory-based and model-based CFs. The memory-based CF refers to an all user-items matrix of a data set for predicting preference scores. It evaluates the similarity between each user or item, generates nearest neighborhoods, and predicts preference scores with nearest neighborhoods. The evaluation of similarity is the most essential step, and the evaluated similarity is used as a weight for predicting preference scores and as a measure for generating nearest neighborhoods [5], [6], [7], [8].

In this paper, we propose novel similarity measures using difference score entropy. The proposed measure evaluates the entropy with the difference of preference scores of common rating items between two users, and normalizes it for use as a weight. Because the proposed similarity measure exploits the advantages of information entropy, evaluation of the proposed similarity measure is simple, and it is possible to change the weight according to score difference of common rating items. Furthermore, the proposed measure shows better prediction accuracy than existing similarity measures. The scope of applications of the proposed similarity measure is vast. For example, it could be used for finding document or image similarities.

The remainder of the paper is organized as follows. We describe various similarity metrics and memory-based CF systems in Section 2. Then, we explain the proposed similarity measure using difference score-based information entropy in Section 3. In Section 4, we present a variety of experimental results for the proposed similarity measure, in terms of the mean absolute error (MAE), number of data sets and each similarity algorithm. Section 5 concludes the paper.

II. RELATED WORKS

In this section, we explain the prediction process of memory-based CF and various similarity metrics. The memory-based CF system considers all records of the data set for predicting preference scores of test users l (or recommendation of the target user). Fig. 1 shows the generic form of the data set for CF

	I_1	I_2	I_3	...	I_n
U_1	r_{U_1,I_1}	r_{U_1,I_2}	r_{U_1,I_3}	...	r_{U_1,I_n}
U_2	r_{U_2,I_1}	r_{U_2,I_2}	r_{U_2,I_3}	...	r_{U_2,I_n}
U_3	r_{U_3,I_1}	r_{U_3,I_2}	r_{U_3,I_3}	...	r_{U_3,I_n}
U_4	r_{U_4,I_1}	r_{U_4,I_2}	r_{U_4,I_3}	...	r_{U_4,I_n}
	\vdots	\vdots	\vdots		\vdots
U_n	r_{U_n,I_1}	r_{U_n,I_2}	r_{U_n,I_3}	...	r_{U_n,I_n}

Figure 1. Form of user-item data set for CF

In Fig. 1, U_i indicates the user, I_j indicates the item and r_{U_i,I_j} indicates the rating of item I_j for user U_i . The real-world generic data set includes several empty ratings. The major aim in CF is to predict the empty ratings. The predicted rating is considered in recommending an item.

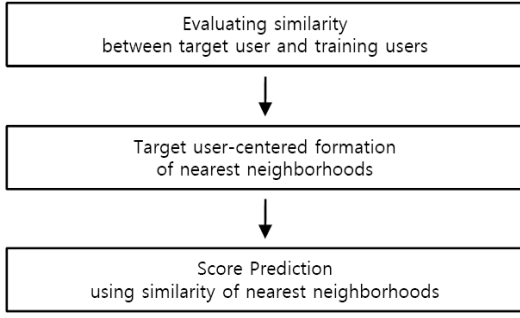


Figure 2. Block diagram of memory-based CF.

The prediction process in memory-based CF contains three steps [5]. They are similarity evaluation, generation of nearest neighborhoods and score prediction. For evaluation of the performance, the CF system considers the mean absolute error (MAE), precision and recall. The CF performance varies according to the processing method of each step. Fig. 1 shows a block diagram of a memory-based CF. paper proposes a novel similarity measure. To this end, we concentrated on existing similarity evaluation methods for memory-based CF.

A. Existing Similarity Measures

The most important first step in memory-based CF is similarity evaluation. The CF system in this step evaluates the similarity between the target user and other users for common rating items. The similarity is used as a weight for predicting the preference score. Various similarity metrics have been proposed in previous studies. These are as follows:

Tanimoto coefficient. It is similarity between two sets [6]. It is a ratio of intersections. Assume that set X is $\{B, C, D\}$ and set Y is $\{C, D, E\}$. The Tanimoto coefficient T of two set A and B is 0.5. This metric doesn't consider the user rating but the case of a very sparse data set is efficient. It is represented by equation (1).

$$T(X, Y) = \frac{X \cap Y}{(X + Y) - (X \cap Y)} \quad (1)$$

Cosine similarity. The Cosine similarity is known as the Vector similarity or Cosine coefficient [7]. This metric assumes that common rating items of two users are two points in a vector space model, and then calculates $\cos\Theta$ between the two points. It is represented by Equation (2).

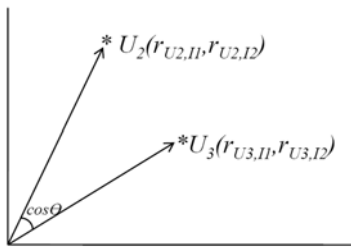


Figure 3. The cosine angle between two points in vector space.

Fig. 3 shows $\cos\Theta$ between two points, for two dimensions. Each common rating item is a dimension. It is well known that the Cosine similarity is very useful for evaluating document similarity in the field of information retrieval. But it is compared with other similarity measures in the CF field.

$$\cos(U_1, U_2) = \frac{\sum_{i=1}^n r_{U_1i} r_{U_2i}}{\|U_1\| \|U_2\|} \quad (2)$$

Person's Correlation. The Pearson Correlation Coefficient r is calculated by Equation (3). In Equation (3), S_{U_j} is the standard deviation of user U_j . The Pearson Correlation measures the strength of the linear relationship between two variables [8], [9], [10], [11]. It is usually signified by r , and has values in the range $[-1.0, 1.0]$. Where -1.0 is a perfect negative correlation, 0.0 is no correlation, and 1.0 is a perfect positive correlation.

$$r(U_1, U_2) = \frac{\sum_{i=1}^n (r_{U_1i} - \bar{U}_1)(r_{U_2i} - \bar{U}_2)}{S_{U_1} S_{U_2}} \quad (3)$$

Spearman's Rank Correlation. The Spearman Rank Correlation also measures the strength of the linear relationship between two variables [5]. Unlike the Pearson Correlation, this metric considers rank of scores. So this similarity measure has more general applicability than the Pearson Correlation, which isn't suitable outside a normalized preference range. Because the range of preference scores for CF is normalized, the Spearman Rank Correlation in the CF field shows comparable performance to the Pearson Correlation. The Spearman Rank Correlation ρ is calculated by Equation (4).

$$r(U_1, U_2) = \frac{6 \sum (rank(r_{U_{1i}}) - rank(r_{U_{2i}}))^2}{n(n^2 - 1)} \quad (4)$$

B. Formation of Nearest Neighbor

The second step after the similarity evaluation is generation of nearest neighborhoods. To improve performance, many methods have been proposed by CF researchers. The methods for selecting nearest neighborhoods include classification using *K-means*, a threshold for the number of common rating items and a graph algorithm. In general, it selects similar users greater than a given threshold or high rank users [10].

C. Prediction of Preference Score

The last step in memory-based CF is to predict the preference score of the target user for non-rating items. It predicts the preference score of non-rating items for the target user, based on the rating of nearest neighborhoods. Various methods have been proposed, and we use the Weighted Mean as most general algorithm [5]. This is represented by Equation (5). PS_{U_i, I_i} is the predicted score of item i for U_i , and NNU_i is the nearest neighbor i .

$$ps_{U_1, I_i} = \frac{\sum_{i=1}^n \text{sim}(U_1, NNU_i) r_{NNU_i, I_i}}{\text{sim}(U_1, NNU_i)} \quad (5)$$

D. Performance Evaluation

In the CF system, there are two types of measure for the performance evaluation. The first type is prediction accuracy, which is evaluated by *MAE* [8], [9], [10], [11]. This is represented by Equation (6). P_i is the real preference score of item i and q_i is the predicted score of item i .

$$MAE = \frac{\sum_{i=1}^n |p_i - q_i|}{n} \quad (6)$$

The second type is recommendation quality, which is evaluated by *precision* and *recall*. The *precision* is the percentage of movies classified as *higher* that are *higher*, and *recall* is the percentage of higher items that were classified as *higher*. In addition to this, the F-measure is also used. The F-measure was proposed as means of intuitively representing the two measures and overcoming the inverse proportion of precision and recall [11]. This is represented by Equation (7). In equation, p is precision and r is recall.

$$F - \text{measure} = \frac{2rp}{r + p} \quad (7)$$

III. SIMILARITY MEASURES USING DEFERENCE ENTROPY

The proposed similarity measure is based on scores of common rating items between two users, and is implemented in the same manner as the existing similarity metric. To explain the proposed similarity measure, assume the user-item matrix such as Table 1 in Section 2. The proposed similarity measure consists of three steps.

Step 1. It evaluates the difference of common rating items between two users. Assume a common rating between $U_1 = \{r_{U_1, I_1}, r_{U_1, I_2}, r_{U_1, I_3}, \dots, r_{U_1, I_n}\}$ and $U_2 = \{r_{U_2, I_1}, r_{U_2, I_2}, r_{U_2, I_3}, \dots, r_{U_2, I_n}\}$. The difference score set $D(U_1, U_2)$ between user U_1 and U_2 is as follows:

$$\begin{aligned} D(U_1, U_2) &= \{r_{U_1, I_1} - r_{U_2, I_1}, r_{U_1, I_2} - r_{U_2, I_2}, \dots, r_{U_1, I_n} - r_{U_2, I_n}\} \\ &= \{d_1, d_2, d_3, \dots, d_n\} \end{aligned}$$

Step 2. It evaluates the weighted difference information entropy $WDE(U_1, U_2)$. When the result is zero, the two users are perfectly similar. On other hand, if the result is higher, the two users aren't similar. For the first time, the information entropy $H(D)$ of D is given by Equation (8) [12]. The $p(d_i)$ is probability density function (PDF) of the difference score between the two users.

$$\begin{aligned} H(D) &= \sum_{i=1}^n p(d_i) \log_2 \left(\frac{1}{p(d_i)} \right) \\ &= - \sum_{i=1}^n p(d_i) \log_2 p(d_i) \end{aligned} \quad (8)$$

When the information entropy $H(D)$ of D is calculated, it can be weighted according to each score difference of common rating items. It can be generalized as follows:

$$WDE(U_1, U_2) = - \sum_{i=1}^n p(d_i) \log_2 p(d_i) \times |d_i| \quad (9)$$

It is further extended by a weighting parameter, such as $d_i^2, d_i^3 \dots d_i^n$ or the square root of d_i . These flexible weighting methods can result in better results than the absolute weighting of Equation (10). This extensional possibility is a major advantage of the proposed similarity measure.

Step 3. It normalizes $WDE(U_1, U_2)$ to $[0, 1]$. The reason this process is needed is that $WDE(U_1, U_2)$ ranges from 0 to infinity. To normalize, we can consider various functions, such as tanh of SVM, and the sigmoid, Fuzzy and Gaussian function. This is the other advantage of the proposed similarity measure. The performance of the CF system using WDE can be improved according to the change in the normalization method. Equation (11), (12) and (13) represents the Gaussian, Sigmoid and tanh function, respectively, where x is the similarity.

$$G(x) = e^{-x^2/2\sigma^2} \quad (10)$$

$$P(x) = \frac{1}{1 + e^{-x}} \quad (11)$$

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (12)$$

Fig. 4 shows the normalized result using the tanh, sigmoid, Gaussian function with the proposed similarity measure.

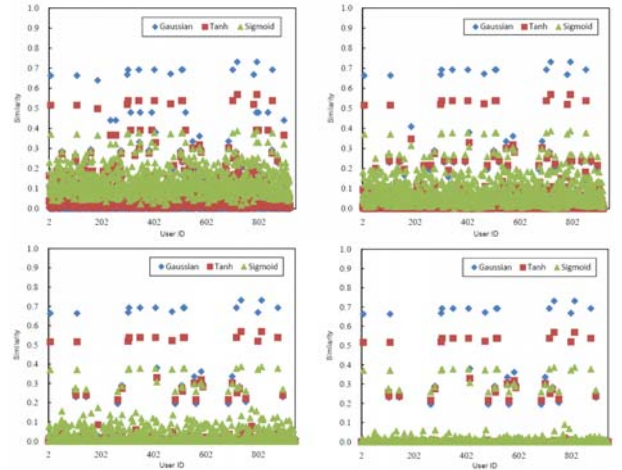


Figure 4. Block diagram of memory-based CF.

It is a result for the user ID 1 of the MovieLens data set. Each normalization method shows a similar rank but the similarity value is different. In Fig. 4, users similar to user ID 1 are clearly shown according to various weightings. Sensible users can be applied to generation of nearest neighborhoods.

We compared the Cosine similarity with the proposed similarity measure, which was weighted as a square and normalized as a Gaussian function, as shown in Fig. 5. The x-axis is the user ID and the y-axis is the similarity. We confirmed that the Cosine similarity is assembled. This result for the Cosine similarity shows that similarity assortment between two users is difficult, and it is ambiguous. The proposed similarity measure shows similar users more clearly than the Cosine similarity and Person Correlation Coefficient.

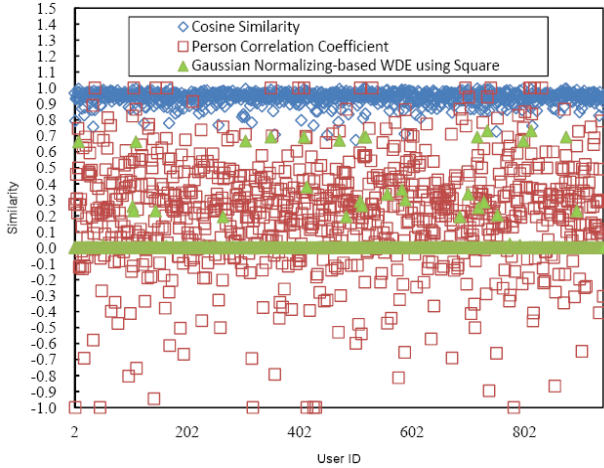


Figure 5. Similarity distribution WDE and existing measures

The algorithm of the proposed similarity measurement is as follows. Because the mechanism is simple, its application is easy. The proposed similarity measure can be applied to various fields. For example, it can be used in comparison between images of the same size, detecting similar signals, document similarity measurement, and evaluating the similarity between two variables. In Section 4, we perform verification via its application to the CF system.

```

BEGIN

// algorithm name: weighted difference entropy
// author: Hyeong-Joon Kwon(katsyuki@skku.edu)

// length of co-rating items
constant length;

// preference array
preference array x[length];
preference array y[length];

// deference set
difference set D[length];

// calculation entropy and weighting
for(int i=1; i<sizeof(x); i++)
    D[i] = abs(x[i] - y[i]);

for(int i=1; i<sizeof(D); i++)
    WDE += p(d[i])*log(p(d[i]),2)*d[i];

// return inverse value as result
return normalization(-WDE);

END

```

IV. EXPERIMENTAL RESULTS

A. Experimental Conditions

We implemented a user-based CF system (one of the memory-based CF systems) for our experiment, using C# (.NET framework). The experiment proceeded in a desktop PC environment consisting of Intel Core 2 Duo E8400 @3.00GHz and 2GB RAM.

We considered the MovieLens data set (GropuLens project) to verify the proposed similarity measure [2]. This data set contains 100,000 ratings for 1,682 movies by 943 users. The rating scale for this data set is from 1 to 5, {1, 2, 3, 4, 5}. The proposed similarity measure was normalized as a Gaussian function for this experiment. The reason is that the inverse function is generally used as a weight in the tanh and sigmoid function. The Gaussian function doesn't need an inverse.

We used the *k*NN method for generation of nearest neighborhoods. To evaluate the performance of the proposed similarity measure, we experimented with various numbers of neighborhoods. Then, we used the Weighted Mean to predict the preference score, as explained in Section 2. This is the most generic method for predicting the score.

B. Performance Evaluation

In this experiment, we randomly extracted 20% of user ratings from the MovieLens data set, and we predicted extracted ratings. And then, we measured the change of MAE according to the number of nearest neighborhoods. MAE decreases until the number of nearest neighborhoods reaches the arbitrary point where it starts to increase.

The Pearson correlation is generally known as one of the most useful metrics in CF, and the performance of the Spearman correlation is also known to be comparable [5]. In Fig. 6, WDE using absolute weighting shows that the improvement of the prediction accuracy was remarkable for the proposed similarity metric, compared with existing similarity metrics.

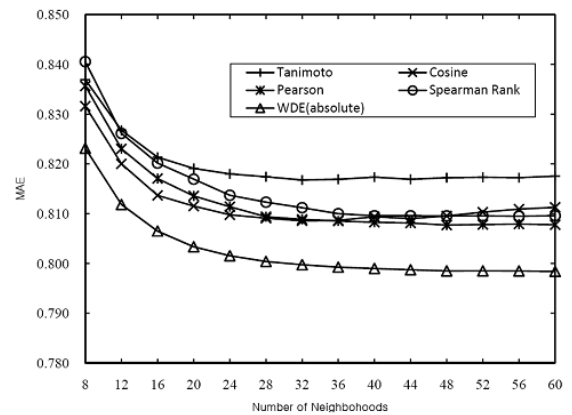


Figure 6. Experimental result for a given number for neighborhoods

In addition, this was suggested by the difference entropy according to Equation (10) in Section 3. The results for the various weighting methods are shown in Fig. 7. We confirmed that changing the weighting

reduces MAE, as expected. But the weighting using a square root increases MAE. Although square root weighting shows bad results, its performance differs according to the format of the data set. For example, the MovieLens data set consists of a range [1, 5]. But the Jester data set is [-10.00, 10.00] and the Book-Crossing data set is [1, 10].

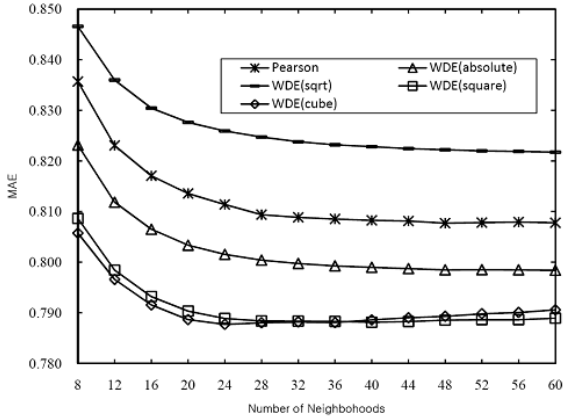


Figure 7. Weighting result for a given number for neighborhoods

The problem of a sparse data set is one of the major issues of CF systems. In case of a data set containing only a few user ratings, the recommendation quality and prediction accuracy is reduced. All CF systems suffer from the sparse data set problem. To evaluate the performance for a sparse data set, we conducted two types of experiments reflecting the differing numbers of ratings available to the recommenders, *All But One* and *Given 5*. In the first type, *All But One*, we extracted a single randomly selected rating for each user in the test set, and tried to predict the given value based on all the other ratings the user submitted. In the second type *Given 5* we randomly selected five ratings from each test user as observed ratings, and then attempted to predict the remaining ratings. This is a similar experimental method to those used in many other studies.

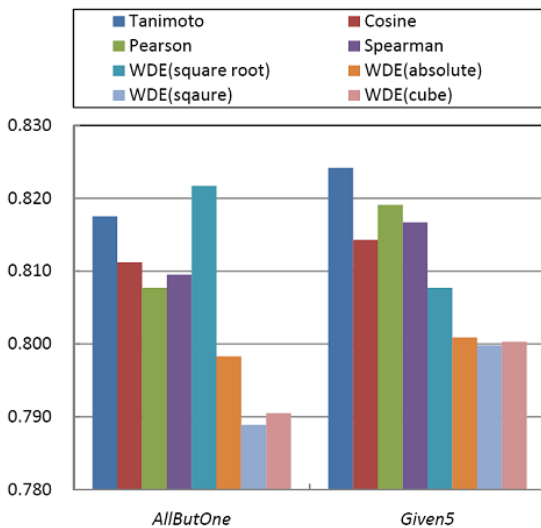


Figure 8. Experimental result for sparse data set

Fig. 8 shows the results for the sparse data experiment. The x-axis is MAE, and the y-axis is the experimental conditions. In Fig. 7, WDE showed better results than existing metrics. The MAE in the CF system was significantly decreased by WDE using various weighting methods. As shown in the results of this experiment, the most important point is that the proposed method showed better performance and prediction accuracy compared to existing methods.

C. Quality of Recommendation

To evaluate the quality of recommendation, various characteristics have been considered, such as ROC-4, precision and recall. The quality of recommendation varies according to the generation method of the nearest neighborhood. It is affected by the similarity metric, but the effect is insignificant. The aim of this experiment was not improvement of the quality of recommendation; the proposed method is about the deterioration of the quality of recommendation. Although the proposed WDE improves the prediction accuracy, if it lowers the quality it will not be useful.

TABLE I.
EXPERIMENTAL RESULT FOR RECOMMENDATION QUALITY

Similarity Measures		Precision	Recall	F-measure
Tanimoto		0.6764	0.2594	0.3749
Cosine		0.6817	0.2612	0.3776
Pearson		0.7039	0.2687	0.3889
Spearman Rank		0.6951	0.2766	0.3957
WDE	\sqrt{d}	0.6703	0.2744	0.3893
	$ d $	0.7021	0.2797	0.4000
	d^2	<u>0.7102</u>	<u>0.2884</u>	<u>0.4102</u>
	d^3	0.7098	0.2810	0.4026

To improve the precision and recall, it must consider the generation of the nearest neighborhood in memory-based CF, because of factors affecting the recommendation quality. Based on Table 1, WDE also improves the quality of recommendation together with MAE. But this result is merely due to the reduction of MAE. The aim of this experiment was to investigate the deterioration of quality, rather than its improvement.

V. CONCLUSIONS

We proposed a novel similarity measure using weighted difference entropy (WDE) to improve the performance of the CF system. The proposed similarity metric evaluates the entropy with a preference score difference between the common rated items of two users, and normalizes it based on the Gaussian, tanh and sigmoid function. We showed significant improvement of experimental results and environments. These experiments involved changing the number of nearest neighborhoods, and we presented experimental results for two data sets with different characteristics, and results for the quality of recommendation.

ACKNOWLEDGMENT

This paper was supported by Samsung Research Fund, Sungkyunkwan University, 2008 and the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. 2008-000-10642-0).

REFERENCES

- [1] Gediminas Adomavicius and Alexander Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. on Knowledge and Data Engineering*, vol. 17, no. 6, June 2005, pp. 734-749.
- [2] Bradley N. Miller, Istvan Albert, Shyong K. Lam, Joseph A. Konstan, "MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System", *ACM 8th Int. Conf. on Intelligent user interfaces*, 2003, pp. 263 – 266.
- [3] Dhruv Gupta, Mark Digiiovanni, Berkeley, Hiro Narita, Ken Goldberg, "Jester 2.0: Collaborative Filtering to Retrieve Jokes", *ACM SIGIR 22nd Int. Conf. on Research in Information Retrieval*, 1999, pp. 333.
- [4] Steve Glassman, "EachMovie Collaborative Filtering Data Set", DEC Research Center, <http://research.compaq.com/SRC/eachmovie/>, 1997.
- [5] Herlocker J.L., Konstan J.A., Borchers A. and Riedl, J., "An Algorithmic Framework for Performing Collaborative Filtering" *ACM SIGIR 22nd Int. Conf. on Research and Development in Information Retrieval*, 1999, pp.230-237.
- [6] Toby S., "Programming Collective Intelligence: Building Smart Web 2.0 Applications", O'reilly, 2007.
- [7] E. Garcia, "Cosine Similarity and Term Weight Tutorial", <http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html#Cosim>. 2006.
- [8] Fu Lee Wang, "Improvements to Collaborative Filtering Algorithm", *Lecture Note in Computer Science*, Vol. 3314, 2004, pp. 975-981.
- [9] Sarwar B., Karypis G., Konstan J. and Riedl J., "Item-Based Collaborative Filtering Recommendation Algorithms", *ACM 10th Int. Conf. on World Wide Web*, 2001, pp.285-295.
- [10] Taek-Hun Kim and Sung-Bong Yang, "An Improved Recommendation Algorithm in Collaborative Filtering", *Lecture Note in Computer Science*, vol. 2455, 2005, pp. 254-261.
- [11] D. Billsus and M.J. Pazzani, "Learning Collaborative Information Filters," *Proc. 15th Int. Conf. Machine Learning*, pp. 46-54, 1998.
- [12] Shannon, Claude E.: Prediction and entropy of printed English, *The Bell System Technical Journal*, Vol. 30, 1951, pp.50-64.