# Applying Consistency-Based Trust Definition to Collaborative Filtering

**Hyoung Do Kim**
School of Business Administration, Hanyang Cyber University
17 HaengDang-Dong, SeongDong-Gu, Seoul 133-791, Rep. of Korea
[e-mail: hdkim@hycu.ac.kr]

*Received June 2, 2009; revised July 20, 2009; accepted July 28, 2009; published August 31, 2009*

## Abstract

In collaborative filtering, many neighbors are needed to improve the quality and stability of the recommendation. The quality may not be good mainly due to the high similarity between two users not guaranteeing the same preference for products considered for recommendation. This paper proposes a consistency definition, rather than similarity, based on information entropy between two users to improve the recommendation. This kind of consistency between two users is then employed as a trust metric in collaborative filtering methods that select neighbors based on the metric. Empirical studies show that such collaborative filtering reduces the number of neighbors required to make the recommendation quality stable. Recommendation quality is also significantly improved.

**Keywords:** Collaborative filtering, consistency, trust, entropy, recommendation

**DOI: 10.3837/tiis.2009.04.002**

## 1. Introduction

Recommender systems assist users to find the information most relevant to their preferences. Collaborative filtering (CF) is one of the most successful technologies used. CF has been developed and improved over the past decade to the point where a wide variety of algorithms exist to generate recommendations [1]. They can be classified into two groups: memory-based and model-based. Memory-based CF uses a similarity measure between pairs of users to build a prediction, typically through a weighted average. The chosen similarity measure determines the accuracy of the prediction. Numerous alternatives have been studied [2]. Some potential drawbacks of memory-based CF include scalability and sensitivity to data sparseness [3]. In general, schemes that rely on similarities across users cannot be pre-computed for fast online queries. Another critical issue is that memory-based schemes must compute a similarity measure between users. There are also many model-based approaches to CF. They are based on linear algebra or on AI techniques, such as neural networks and clustering [4]. Model-based CF algorithms are typically faster at query time than memory-based schemes, though they might have expensive learning or updating phases. Model-based schemes can be preferable to memory-based schemes when query speed is crucial.

Memory-based CF either uses neighbors of users or of items to compute a prediction. The first kind is called user-based, and the second item-based. They form neighborhoods by computing the similarity between all pairs of users or items. Predictions are then computed by aggregating ratings; in a user-based algorithm, this involves aggregating the ratings of items that are neighbors of the target item. Algorithms within these families differ in the definition of similarity, formation of neighborhoods, and the computation of predictions.

User-based CF has been explored in-depth during the last ten years and represents the most popular recommendation algorithm, owing to its compelling simplicity and excellent quality of recommendations [5][6][7]. User-based CF recommends items by building the customer profiles from their preferences for each item. Preferences are generally represented as numeric values rated by customers. Predicting a preference for a certain product that is new to the test customer is based on the ratings of other customers for the target item. Therefore, it is important to find a set of customers with similar preferences to the test customer for better prediction quality.

CF operates on a set of users $U = \{u_1, \ldots, u_n\}$ and a set of products $P = \{p_1, \ldots, p_m\}$. In user-based CF, equation (1) is used to predict customer preferences by the typical Pearson correlation coefficient to calculate similarity.

$$p_{u,i} = \bar{r}_u + \frac{\sum_v \{sim(u,v) \times (r_{v,i} - \bar{r}_v)\}}{\sum_v |sim(u,v)|} \quad where \;\; sim(u,v) = \frac{\sum_j (r_{u,j} - \bar{r}_u)(r_{v,j} - \bar{r}_v)}{\sqrt{\sum_j (r_{u,j} - \bar{r}_u)^2 \sum_j (r_{v,j} - \bar{r}_v)^2}} \tag{1}$$

In the above equation, $p_{u,i}$ is the preference of the test customer $u$ with respect to the target item $i$. $r_{v,i}$ and $r_{v,j}$ are customer $v$'s ratings for items $i$ and $j$, respectively, and $r_{u,j}$ is customer $u$'s rating for item $j$. $r_u$ and $r_v$ are the averages of customer $u$'s ratings and customer $v$'s ratings, respectively.

If customer $u$ and $v$ have similar ratings for an item, then $sim(u,v)>0$. $|sim(u,v)|$ indicates how much customer $u$ tends to agree with customer $v$ on the items that both customers have already rated. If they have opposite ratings for an item, then *sim(u,v)*<0 and |*sim(u,v)*| indicates how much they tend to disagree on the item rated by both. Hence, if they do not correlate with each other, then *sim(u,v)*=0. Note that *sim(u,v)* can be in beween -1 and 1 inclusive.

Recommender systems have typically been evaluated using measures of predictive accuracy [2]. They compute predictive accuracy by dividing a set of ratings into training and test sets, and compute the prediction for an item in the test set using the ratings in the training set. A standard measure of predictive accuracy is mean absolute error (MAE) [8].

Since CF is based on the ratings of the neighbors who have similar preferences, it is very important to select the neighbors properly to improve the prediction quality. **Fig. 1** shows that at least 100 neighbors are needed to obtain the stable quality measured in MAE, when nearest neighbors are determined using the Pearson correlation coefficient [9]. Conversely, too many neighbors, used to improve the prediction quality and stability, cause the system performance to degrade. The stabilty may not be so good. This means that high similarity between any two users does not guarantee the same preference for some products considered for recommendation. The smoothing effect of increasing the number of neighbors overrules preference prediction.

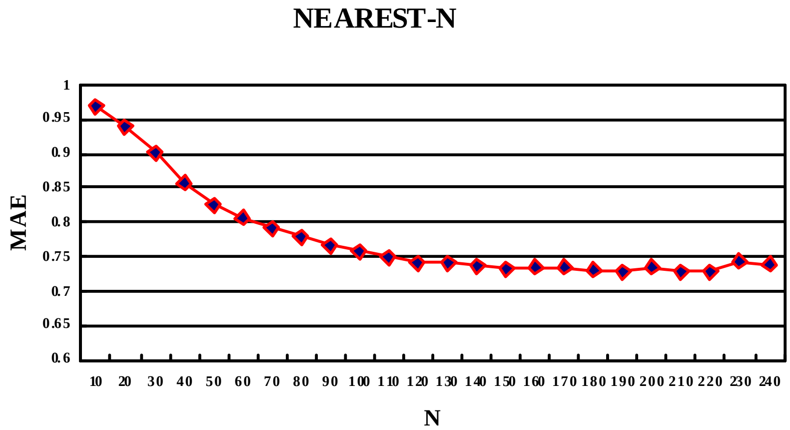


**Fig. 1**. Prediction Quality of NEAREST-N in MAE [9]

In real-world prediction, the consistency of a user's behavior is more important than similarity with those of other users. For instance, the ratings of a user *A* for products are opposite with those of another user *B*, this characteristic of user *B* can be employed to predict user *A*'s preferences. When determining neighbors based on similarity, user *B* cannot be a neighbor of user *A*. The starting point of this paper was the need to consider consistency as a practical metric of CF rather than similarity. This kind of consistency can be extended to the definition of trust between users. According to Webster's dictionary, trust means assured reliance on the character, ability, strength, or truth of someone or something. In this paper, trust is interpreted as an agent's expectation of another agent's competency in providing opinions to reduce its uncertainty (i.e. inconsistency) in predicting new items' ratings.

Note that early recommender systems were said to simulate the informal, verbal exchange of information known as word of mouth communication [10]. Recommender systems produce social networks [11], i.e. user-to-user connections, as a consequence of predicting user-to-item connections [12]. Within such a social network, user trustworthiness is an important factor

[13]. Only recently has trust been introduced into CF. Massa and Avesani [14] proposed that a peer can establish trust with other peers through explicit trust statements and trust propagation. However, it is not clear how to quantify the degrees of trust when making trust statements. Papagelis et al. [15] developed a model to establish trust between users by exploiting the transitive nature of the trust. However, it simply adopts similarity for trustworthiness. Hence, it still possesses the limitations of similarity-based CF. Weng et al. [16] designed a trust metric that helps a user to quantify the degree of trust it should place on a particular user. The trust metric is computable on most users, even on pairs of users who have only co-rated one common item. However, neighbors cannot be changed during the prediction process. This paper presents how to define trust measured by entropy-based consistency to overcome these limitations.

The remainder of this paper is organized as follows. Section 2 presents how to define trust and consistency based on entropy. Section 3 explains alternative ways to predict preference using trust. Section 4 discusses the meaning of experimental results using an example data set. Finally, section 5 concludes the paper with some future research directions.

## 2. Consistency-Based Definition of Trust

Entropy is used as a means to measure uncertainty related to a random variable in information theory. It has been applied to data compression, establishing decision tree etc. because entropy can quantify information contained in data. The entropy of a discrete random variable, which can take values of $\{x_1, \ldots x_M\}$, can be expressed as equation (1). If the probability of M=2 and $X=x_1$ is p, then the entropy can be expressed as equation (2). When p=0.5, the binary entropy function gains its maximimum value. This means that uncertainty is maximal when p=0.5.

$$E(X) = -\sum_{i=1}^{M} \Pr(X = x_i)\log_2 \Pr(X = x_i) \quad \text{s.t.} \sum_{i=1}^{M} \Pr(X = x_i) = 1 \tag{1}$$

$$E(p) = -p\log_2 p - (1-p)\log_2(1-p) \tag{2}$$

The entropy of an active user can be calculated by equation (3) based on the number of times of rated values contained in the past data. Here Z denotes the number of states of rated values, N is the total number of rating times. As in **Table 1**, Z=5 and N=10 when 10 ratings are made with 1 to 5 integer-valued scores.

$$Entropy = -\sum_{i=1}^{Z} \frac{n_i}{N} \log \frac{n_i}{N} \tag{3}$$

If the neighbors of an active user are already determined, entropy could be calculated more delicately. The number of rating times can be summarized as the table form of **Fig. 2**. Here, u is the active user and v a neighbor. The entropy of this state is calculated by equation (4). As a result, entropy improvement, by obtaining a neighbor, is determined by subtracting equation (4) from equation (3). Neighbors can be selected using this kind of entropy improvement for better preference prediction.

However, we need to rearrange equation (3), since equations (3) and (4) are not in the same dimension. In order to make it possible, we assume that the numbers of rating times for each value are distributed evenly on all the rated values of an imaginary neighbor, as in **Fig. 3**. The entropy of this state is calculated by equation (5).

**Table 1**. Meaning of Z and N

| Rated value<br>i | Number of rating times<br>$n_i$ | Ratio<br>$n_i/N$ |
|---|---|---|
| 1 | $n_1=2$ | $n_1/N$ |
| 2 | $n_2=1$ | $n_2/N$ |
| 3 | $n_3=3$ | $n_3/N$ |
| 4 | $n_4=0$ | $n_4/N$ |
| 5=Z | $n_5=4$ | $n_5/N$ |
| summation | $N=n_1+n_2+n_3+n_4+n_5=10$ | 1 |

Ratings by *v*

Ratings by *u*

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | $n_{11}$ | $n_{12}$ | $n_{13}$ | $n_{14}$ | $n_{15}$ |
| 2 | $n_{21}$ | $n_{22}$ | $n_{23}$ | $n_{24}$ | $n_{25}$ |
| 3 | $n_{31}$ | $n_{32}$ | $n_{33}$ | $n_{34}$ | $n_{35}$ |
| 4 | $n_{41}$ | $n_{42}$ | $n_{43}$ | $n_{44}$ | $n_{45}$ |
| 5=Z | $n_{51}$ | $n_{52}$ | $n_{53}$ | $n_{54}$ | $n_{55}$ |

**Fig. 2**. Number of times given a neighbor

Imaginary Neighbor's Ratings

Ratings by *u*

| | 1 | 2 | 3 | 4 | 5 | Total |
|---|---|---|---|---|---|---|
| 1 | $n_1/Z$ | $n_1/Z$ | $n_1/Z$ | $n_1/Z$ | $n_1/Z$ | $n_1$ |
| 2 | $n_2/Z$ | $n_2/Z$ | $n_2/Z$ | $n_2/$ | $n_2/Z$ | $n_2$ |
| 3 | $n_3/Z$ | $n_3/Z$ | $n_3/Z$ | $n_3/Z$ | $n_3/Z$ | $n_3$ |
| 4 | $n_4/Z$ | $n_4/Z$ | $n_4/Z$ | $n_4/Z$ | $n_4/Z$ | $n_4$ |
| 5=Z | $n_5/Z$ | $n_5/Z$ | $n_5/Z$ | $n_5/Z$ | $n_5/Z$ | $n_5$ |

**Fig. 3**. Number of times given no neighbors

$$
\begin{aligned}
&Entropy(u \mid v) \\
&= -\sum_{i=1}^{Z}\sum_{j=1}^{Z}\frac{n_{ij}}{N}\log\frac{n_{ij}}{N} = -\sum_{i=1}^{Z}\sum_{j=1}^{Z}\frac{n_{ij}}{N}(\log n_{ij} - \log N) = -\sum_{i=1}^{Z}\sum_{j=1}^{Z}\frac{n_{ij}}{N}\log n_{ij} + \sum_{i=1}^{Z}\sum_{j=1}^{Z}\frac{n_{ij}}{N}\log N \\
&= \log N - \frac{1}{N}\sum_{i=1}^{Z}\sum_{j=1}^{Z} n_{ij}\log n_{ij} \qquad (4)
\end{aligned}
$$

$$
\begin{aligned}
&Entropy(u) \\
&= -\sum_{i=1}^{Z}\sum_{j=1}^{Z}\frac{n_i}{ZN}\log\frac{n_i}{ZN} = -\sum_{i=1}^{Z}\frac{Zn_i}{ZN}(\log n_i - \log ZN) = -\sum_{i=1}^{Z}\frac{n_i}{N}\log n_i + \sum_{i=1}^{Z}\frac{n_i}{N}\log ZN \\
&= \log ZN - \frac{1}{N}\sum_{i=1}^{Z} n_i \log n_i
\end{aligned} \tag{5}
$$

This kind of entropy definition between two users means that the entropy value becomes lower as their ratings are more consistent. Similar to the definition of trust, as the expectation of technically competent role performance [17], trust can be interpreted as an user's expectation of another user's competence in providing opinions to reduce its uncertainty in predicting new item ratings. The improvement ratio of entropy by associating a user with a neighbor is then definable as trust, as in equation (6).

$$
\begin{aligned}
Trust(u \to v) &= \frac{Entropy(u) - Entropy(u\,|\,v)}{Entropy(u)} = \frac{(\log ZN - \frac{1}{N}\sum_{i=1}^{Z} n_i \log\ n_i) - (\log N - \frac{1}{N}\sum_{i=1}^{Z}\sum_{j=1}^{Z} n_{ij} \log\ n_{ij})}{\log ZN - \frac{1}{N}\sum_{i=1}^{Z} n_i \log\ n_i} \\
&= \frac{(1 - \frac{1}{N\log ZN}\sum_{i=1}^{Z} n_i \log\ n_i) - (\frac{\log N}{\log ZN} - \frac{1}{N\log ZN}\sum_{i=1}^{Z}\sum_{j=1}^{Z} n_{ij}\log\ n_{ij})}{1 - \frac{1}{N\log ZN}\sum_{i=1}^{Z} n_i \log\ n_i} \\
&= \frac{(1 - \frac{\log N}{\log ZN}) + \frac{1}{N\log ZN}(\sum_{i=1}^{Z}\sum_{j=1}^{Z} n_{ij}\log\ n_{ij} - \sum_{i=1}^{Z} n_i \log\ n_i)}{1 - \frac{1}{N\log ZN}\sum_{i=1}^{Z} n_i \log\ n_i}
\end{aligned} \tag{6}
$$

## 3. Prediction Methods

We can predict the user's preferences based on the past ratings of neighbors consistent with the active user. The simplest method is to select the best trusty neighbor of the active user, whose ratings in the past are reused as is. This can be expressed as equation (7), where $r_{v,,x}$ is the rating value of a neighbor *v* of the active user *u* on product *x* and $p_{u,x}$ is the preference value predicted on the active user. This method is termed NAIVE.

$$
\begin{gathered}
NAIVE \\
p_{u,x}\,|\,r_{v,x} = r_{v,x}
\end{gathered} \tag{7}
$$

Another method calculates the weighted average of the active user's ratings for all the cases where a neighbor v rated any product as $r_{v,,x}$. This method is termed HISTORIC.

Another method reflects the differences of the past ratings as equation (9), where $r_{v,,x}$ is added to the average difference of two rated values k and l. This is termed AVGDIFF.

Any combined methods, such as HISTORIC and NAÏVE, or HISTORIC and AVGDIFF can also be applied to collaborative filtering.

*HISTORIC*

$$p_{u,x} \mid r_{v,x} = \sum_{k=1}^{Z} k \cdot n_{k,r_{v,x}} \Big/ \sum_{k=1}^{Z} n_{k,r_{v,x}} \quad \text{if} \sum_{k=1}^{Z} n_{k,r_{v,x}} \neq 0 \tag{8}$$

*AVGDIFF*

$$p_{u,x} \mid r_{v,x} = r_{v,x} + \sum_{k=1}^{Z}\sum_{l=1}^{Z}(k-l)\cdot n_{k,l} \Big/ \sum_{k=1}^{Z}\sum_{l=1}^{Z} n_{k,l} \cong r_{v,x} + \overline{r_u} - \overline{r_v} \tag{9}$$

As in the case of similarity-based predictions, it is possible to obtain better results by combining the values calculated for selected neighbors. We can use a smoothing method, as equation (10) shows.

$$P_{u,x} = \frac{\sum_{v \in Neighbor\,(u)} Trust\,(u \to v) \cdot p_{u,x} \mid r_{v,x}}{\sum_{v \in Neighbor\,(u)} Trust\,(u \to v)} \tag{10}$$

$$v \in Neighbor\,(u),$$
$$\text{if } \; Trust\,(u \to v) > \theta$$

One advantage of trust-based prediction methods is to use trust propagation. If a neighbor *v*, selected by the active user *u* as trusty, does not have any rating experience for the product to be predicted, a trusty neighbor *w* of user *v* can also be used for the prediction, although users u and w do not have any direct relations. Although there could be diverse kinds of propagation methods, this paper proposes three methods: p-NAÏVE (equation 11), p-AVGDIFF (equation 12), and p_REESTIMATE (equation 13). p-NAIVE just reuses $p_{v,x}$ calculated from the ratings of v's neighbors. Additionally, p-AVGDIFF considers the average difference of ratings of two users *u* and *v* for p-NAIVE. p_REESTIMATE is the weighted average of two estimates calculated using the floor and ceiling values of $p_{v,x}$ in order to consider $p_{v,x}$ is not an integer.

*p−NAIVE*

$$p_{u,x} \mid p_{v,x} = p_{v,x} \tag{11}$$

*p−AVGDIFF*

$$p_{u,x} \mid p_{v,x} = p_{v,x} + \overline{r_u} - \overline{r_v} \tag{12}$$

*p−REESTIMATE*

$$p_{u,x} \mid p_{v,x} = [p_{u,x} \mid floor(p_{v,x})] \times [p_{v,x} - floor(p_{v,x})] + [p_{u,x} \mid ceil(p_{v,x})] \times [ceil(p_{v,x}) - p_{v,x}] \tag{13}$$

## 4. Experimental Results

The methods for collaborative filtering by consistency-based trust definition have been simulated based on the MovieLens dataset [18]. The dataset contains 100,000 ratings of 1682 movies rated by 943 users. The ratings were divided into two groups to evaluate the quality of the prediction measurement: 90% of the data (90,000 ratings) was used as a training set and 10% of the data (10,000 ratings) was used as a test set. The Mean Absolute Error (MAE), computed by equation (14), is used as the statistical accuracy metric to evaluate the mechanism. In the equation, $N$ is the number of predictions and $e_i$ is the error between the predicted rating and the actual rating for product item $i$.

$$MAE = \frac{\sum_{i=0}^{N} |e_i|}{N} \tag{14}$$

Comparing HISTORIC with NEAREST-N, as in **Fig. 4**, the number of neighbors of the former required to obtain the stable MAE value 0.75 is much lower than that of the latter. 5 to 7 neighbors in HISTORIC are sufficient to reach a stable state, whilst almost 100 neighbors are required in NEAREST-N. This explains the efficiency of HISTORIC.

HISTORIC+NAÏVE is the combination of HISTORIC and NAÏVE, where HISTORIC is used just when there is a rating record for the product by the neighbor. NAÏVE is used otherwise. The experimental results of the two methods do not differ much, as **Fig. 4** describes. Trust (consistency) propagation can be employed to compensate, when there is no rating record for the product by the neighbor. **Fig. 4** shows the combination of HISTORIC and PROPAGATION by AVGDIFF has slighlty better MAE than HISTORIC+NAÏVE in the range of 1 to 10 neighbors. Conversely, MAE of HISTORIC+PROPAGATION(AVGDIFF) deteriorates when there are more than 10 neighbors.

In the experiment, HISTORIC+PROPAGATION(REESTIMATE) exhibit the best result. The MAE values go below 0.7 when there are more than 20 neighbors. That is, more than 6 % improvement is possible by employing the HISTORIC+PROPAGATION (REESTIMATE) method. This level of MAE is evaluated as competitive, not only compared to the other prediction methods proposed in this paper, but also compared to other research results [2][19][20] using the dataset for neighbor-based recommendation.

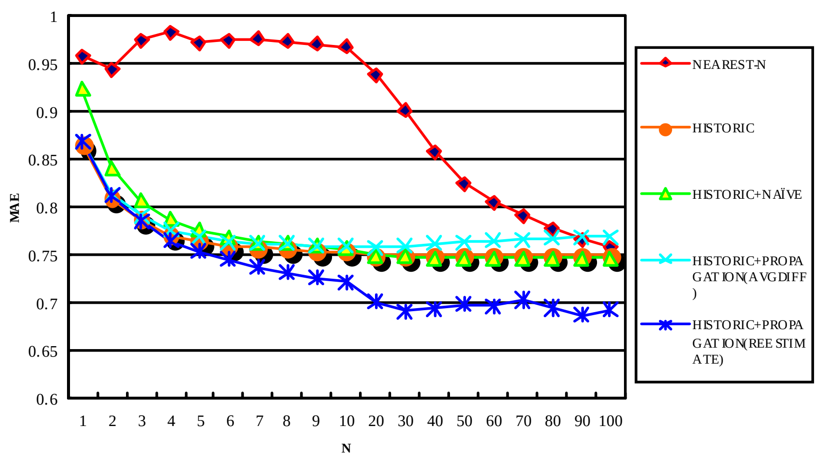


**Fig. 4**. Comparison of Prediction Methods

## 5. Conclusions and Future Directions

This paper presented the definition of consistency using entropy between two users. Such consistency between two users is then employed as a trust metric in collaborative filtering methods that select neighbors based on the metric. Only a small number of neighbors is required to stabilize the recommendation quality. Recommendation quality is also very good. Furthermore, trust (consistency) propagation reduces the severity of the sparsity problem intrinsic to collaborative filtering methods.

Further research needs to be directed toward multi-dimensional trust (consistency) propagation and distributed collaborative filtering methods. Better results are expected from applying more than single-level propagation. Distributed collaborative filtering, appearing as an alternative to centrally-focused collaborative filtering, will distribute workloads to respective agents with enhanced security facilities.

## References

[1] J.L. Herlocker, J.A. Konstan, L. Terveen, and J. Riedl, “Evaluating Collaborative Filtering Recommender Systems,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5-53, 2004.

[2] J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, “An Algorithmic Framework for Performing Collaborative Filtering,” In *Proceedings of the 22nd Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp. 230-237, Berkeley, CA, USA, Aug. 1999.

[3] D. Lemire and A. Maclachlan, “Slope One Predictors for Online Rating-Based Collaborative Filtering,” In *Proceedings of the 2005 SIAM Int’l Conf. on Data Mining (SDM’05)*, Newport Beach, CA, USA, April 2005.

[4] H.-N. Kim, A.-T. Ji, H.-J. Kim, and G.-S. Jo, “Error-Based Collaboration Filtering Algorithm for Top-N Recommendation,” In *Proceedings of APWeb 2007 and WAIM 2007 (LNCS 4505)*, pp. 594-605, Huang Shan, China, June 2007.

[5] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, J. Riedl, and H. Volume, “GroupLens: Applying Collaborative Filtering to Usenet News,” *Communications of the ACM*, vol. 40, no. 3, pp. 77-87, 1997.

[6] B. Sarwar, G. Karypis, J.A. Konstan, and J. Riedl, “Analysis of Recommendation Algorithms for e-Commerce,” In *Proceedings of the 2nd ACM Conf. on Electronic Commerce*, pp. 158-167, Minneapolis, MN, USA, Oct. 2000.

[7] B.K. Mohan, B.J. Keller, and N. Ramakrishnan, “Scouts, Promoters, and Connectors: The Roles of Ratings in Nearest Neighbor Collaborative Filtering,” In *Proceedings of EC’06*, pp. 250-259, Ann Arbor, Michigan, USA, June 2006.

[8] G. Carenini and R. Sharma, “Exploring More Realistic Evaluation Measures for Collaborative Filtering,” In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004)*, San Jose, CA, USA, July 2004.

[9] H.D. Kim, “Collaborative Filtering by Consistency-Based Trust Definition,” In *Proceedings of the 2007 Fall International Conference on Industrialization of Ubiquitous Technology and Balanced National Development*, pp. 551-556, CheongJu, ChungCheongBuk-Do, South Korea, Nov. 2007, (Written in Korean).

[10] U. Shardanand and P. Maes, “Social Information Filtering: Algorithms for Automating ‘Word of Mouth’,” In *Proceedings of the ACM CHI Conf. on Human Factors in Computing Systems*, pp. 210-217, Denver, Colorado, USA, May 1995.

[11] S. Wasserman and K. Faust, Social Network Analysis: Methods and Applications, Cambridge University Press, 1994.

[12] A. Webster and J. Vassileva, “Push-Poll Recommender System: Supporting Word of Mouth,” *LNAI*, vol. 4511, pp. 278-287, 2007.
[13] J. O’Donovan and B. Smyth, “Trust in Recommender Systems,” In *Proceedings of 2005 Int’l Conf. on Intelligent User Interfaces (IUI’05)*, pp. 167-174, San Diego, CA, USA, Jan. 2005.
[14] P. Massa and P. Avesani, “Trust-Aware Collaborative Filtering for Recommendater Systems,” *LNCS*, vol. 3290, pp. 492-508, 2004.
[15] M. Papagelis, D. Plexousakis, and T. Kutsuras, “Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inference,” In *Proceedings of iTrust 2005*, pp. 224-239, Rocquencourt, France, May 2005.
[16] J. Weng, C. Miao, and A. Goh, “Trust-Based Agent Community for Collaborative Recommendation,” In *Proceedings of AAMAS’06*, pp. 1260-1262, Hakodate, Hokkaaido, Japan, May 2006.
[17] B. Barber, The Logic and Limits of Trust, Rutgers University Press, 1983.
[18] GroupLens Research, “MovieLens Data Sets,” http://www.grouplens.org/node/73, 2006.
[19] H.J. Ahn, “A New Similarity Measure for Collaborative Filtering to Alleviate the New User Cold-Starting Problem,” *Information Sciences*, Vol. 128, pp. 37-51, 2008.
[20] H.C. Lee, S.J. Lee, and Y.J. Chung “A Study on the Improved Collaborative Filtering Algorithm for Recommender System,” In *Proceedings of the 5th International Conference on Software Engineering Research, Management and Applications (SERA2007)*, pp. 297-304, Busan, Korea, Aug. 2007.

**HyoungDo Kim** is an associate professor of the School of Business Administration at the Hanyang Cyber University. He received his Ph.D. and Master’s degrees from the Department of Management Science, Korea Advanced Institute of Science and Technology (KAIST), South Korea. Prof. Kim received his BA from the Department of Industrial Engineering, Seoul National University, South Korea. For more than 6 years, he worked for a telecommunication company in the research of EC (Electronic Commerce) and Internet services. As a leader, he has been taking part in Korean e-business standardization efforts, including Korea ebXML committee and ECIF (Integrated Forum for Electronic Commerce). He has published his research papers in international refereed journals such as Decision Support Systems, ACM SIGMOD Record, IEICE Transactions on Information & Systems, Expert Systems and Applications, and so on. His research interests are directed toward data mining (business intelligence), electronic commerce, business processes and workflow systems, digital watermarking, and e-learning (on-line education).