Winter 12-5-2004

A User-Oriented Contents Recommendation System in Peer-to-Peer Architecture

Hyeakyeong Kim

Jae Kyeong Kim

A User-Oriented Contents Recommendation System in Peer-to-Peer Architecture

Hyeakyeong Kim, Jaekyeong Kim

School of Business Administration, Kyung Hee University, Seoul, 130-701, Korea
nell10@korea.com, jaek@khu.ac.kr

ABSTRACT

Recommender system is a popular technique for reducing information overload and finding digital contents that is most valuable to users. However, most recommender systems are based on a centralized client-server architecture in which servers and clients represents contents providers and users respectively. The existing recommender systems depend on contents providers and give a number of disadvantages to users. Therefore, we propose a recommender system based on a distributed P2P architecture that has originated with user-oriented principle rather than business itself. The proposed system consists of fully functioning personal recommender agents that automatically select neighbors and recommend contents. The agents learn user preference from users' content usage without requiring users' explicit ratings. We believe that the suggested P2P based recommender system should provide the users with more qualified recommendations, while it reduces the effort and time of users.

***Keywords**: Recommendation, Peer-to-Peer, Collaborative Filtering, Content-Based Image Retrieval*

1. INTRODUCTION

According to a recent report, 93% of information produced worldwide is in digital form and the unique data added each year exceeds one exabyte, and more than 513 million people around the world are now connected to the global information resource [11]. However, all of those people have problems to search for digital contents they are most interested in. This trend calls for equally recommender systems with scalable searching capability. Recommender systems have been proved to be one of the most successful techniques to help people find contents that are most valuable to them in research and practice.

But, most existing Recommender systems are based on centralized client-server architecture in which servers and clients represent service providers and users respectively. The existing recommender systems depend on content providers and give a number of disadvantages to users. Centralized recommender systems collect users' sensitive information at one server. This makes recommender systems a serious risk for violating the privacy of users. Furthermore, from the users' perspective, the information about their preference is fragmented across many service providers reducing the quality of recommendations to them. Besides, each service provider requires users to indicate their specific preferences. So users must provide redundant information to obtain suitable recommendations from each service provider.

One solution to the problems with centralized recommender system is to distributed peer-to-peer (P2P) architecture, recent alternative to the dominating client-server architecture. Users in P2P architecture works as servers and clients simultaneously.

In this paper, we propose a collaborative filtering-based image content recommender system in distributed P2P architecture as a solution to the problems with centralized recommender system. Our system consists of users connected by personal recommender agents which reside on each user's computer. All processing to recommend contents is done locally by the agent. It allows users to maintain a fraction of the information about other users only, not all users' information, and generates recommendations themselves. Our proposed recommender agent selects more similar neighbors dynamically by learning preference, what image the user want, from the user's content usage without requiring user's explicit ratings. The system combines the two most popular information filtering techniques: Collaborative Filtering and Content-based Image Retrieval. This paper describes the reasons for and by which these two techniques were combined to recommend images in P2P architecture.

2. RELATED WORK

2.1 CF-based Recommender Systems

The recommender system is one of possible solutions to searching for individually preferred images from amount of images. A recommender system is defined as a system that assists customers in finding the items they would like to purchase. One of the most successful recommendation techniques is Collaborative Filtering (CF) [1,3,12,13], which has been widely used in a number of different applications.

Collaborative filtering is an information filtering technique that depends on human beings' evaluations of items. It is an attempt to automate the "word of mouth" recommendations. It identifies customers whose tastes are similar to those of a given customer and it recommends items those customers have liked in the past. In general, CF-based recommender systems make recommendations according to the following steps [3,12]: (1) A customer provides the system with preference ratings on items that may be used to build a customer profile. (2) The system applies statistical or machine learning techniques to find a set of customers, known as neighbors, who had in the past exhibited similar behaviors. A neighborhood is formed based on the degree of similarity between a target customer and other customers. (3) Once a neighborhood is formed for a target customer, the system generates a set of items that the target customer is most likely to purchase by analyzing the items in which neighbors have shown an interest.

Despite their success, none of the previously proposed approaches can be an adequate solution for image recommendations in distributed P2P architecture. In this paper, Content-based Image Retrieval (CBIR) [5,6,10], the most widely used image retrieval technique, was used as a technique for content-based similarity search and was combined with CF.

2.2 Content-Based Image Retrieval

Content-based image retrieval represents an image as a point in multi-dimensional feature space and performs similarity-based retrieval using its visual features such as color, texture and shape. In CBIR, the customer describes visual characteristics of desired images using a query that is a set of example images. A query is internally represented as multiple points (i.e. query points) that have visual characteristics of example images. In general, CBIR systems retrieve images according to the following steps [15]: (1) A customer presents a query to the system via selection or sketching of images as a request for desired images. (2) The system searches for images similar to the query. The similarity between an image in the database and a query is calculated using the distance between corresponding points in the feature space. (3) The images with the highest degree of similarity are retrieved and recommended to the customer.

In spite of the virtues of CBIR in retrieving images similar to a query, CBIR rarely brings a customer to the desired images immediately. The reason for this is that any combination of example images may not precisely represent the images that a customer desires. For a system to handle this gap properly, it needs the ability to learn about what image the customer really wants through iterative interactions. The customer's current preference on the presented images needs to be fed back so that CBIR can learn from this preference to retrieve, in the next iteration, images more similar to the one customer really wants. This learning process, the *preference feedback*, is an essential mechanism for a faster search of desired images. We will refer to a set of preferred images as a *saved image set*. The images in the saved image set are used for query refinement for the purpose of learning customer's current preference.

2.3 P2P (Peer-to-Peer) System

With the pervasive deployment of computers, P2P is increasingly receiving attention in research and practice. P2P system is direct communication or collaboration between two or more agents, such as personal computers or devices that bypass a centralized computer server. Current P2P applications can be classified into one of the following three categories [4]: Content File sharing, Distributed processing, Instant messaging

As the main usage of P2P systems are to share content files among a group of computers called peers in a distributed way, by direct exchange between peers. The using contents in P2P architecture differs markedly from the client-server architecture. A client-server based system depends on a single server storing information and distributing it to clients. The information repository remains essentially static, centralized at the server, and subject only to updates by the provider. Peers assume a passive role in that they receive, but do not contribute, information. A P2P system, on the other hand, considers all peers equal in their capacity for sharing information with other users. Each peer makes an information repository available for distribution, which, combined with anyone's ability to join the network, each peer can make information available for distribution and can establish direct connections with any other peers to download information. Instead of looking at what is available in a centralized repository. a peer seeking information from a P2P system searches across scattered collections stored at numerous peers, all of which appear to be a single repository with a single index. P2P system can use protocols that make it easier for each peer participate and share information, but the trade-off can be decreased quality of service.

Any P2P system doesn't provide recommendations to peers so that the peers have problems to search for contents that are most valuable to peers. To add value to P2P system, it needs to provide efficient search to peers to reduce information overload. We believe that P2P system with recommender functions make it possible. Therefore we propose contents recommender system in P2P architecture[2,8,9].

3. PROPOSED RECOMMENDER SYSTEM

Our system is designed to deal with the problems we face in recommending image files in a decentralized domain, P2P architecture. All processing to support a peer in finding desired images by generating personalized recommendations of image is done locally by a personal agent resides on each peer's computer without central server.

The system consists of interconnected peers of agents are implemented to be fully functioning recommenders. The agent keeps watch its peer (called the ***host peer***) and learn what images the host peer find relevant. That knowledge is then used to find both other relevant images and similar peers (called the ***neighbor set***). The agent exchanges recommendations with the agents of its neighbor set.

The personal agent consists of two modules: CF agent and CBIR agent. Two components collaborate each other to decide whom should receive a recommendation, whom to keep as a neighbor, and what contents relevance to the host peer. Figure 1 shows proposed recommender agent with overall procedure.

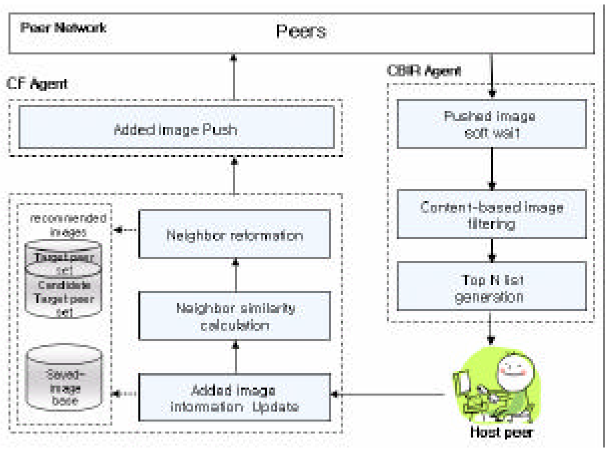

Figure 1. Personal Recommender Agent

The agent dynamically selects which peers to include its neighbor set. When a peer is discovered to frequently provide good recommendations, the peer attempts to move closer to the host peer. Thus an agent might keep better constituent neighbors and recommendations should be reached to more related peers. This mechanism provides a wide peer network with efficient image sharing being spread faster with less communications.

3.1 Peer Model

In P2P architecture, a personal recommender agent builds a peer model respectively to generate personalized recommendations for a host peer. The peer model should include information about a host peer's preference for image and its neighbors to exchange recommendations with other agents.

Image representation: To analyze peers' preference for images, we use the HSV (i.e. hue, saturation, and value of color) based color moment over other choices of features such as shape or texture, because color moment is the most generally used feature and HSV represents human color perception more uniformly than others [9]. For all pixels in images, we translated the values of three-color channels (i.e. RGB: red, green, and blue) into HSV values. Then, the mean, standard deviation and skewness for HSV values were calculated to represent images as vectors in a nine dimensional feature space.

Peer Profile: A peer profile consists of two parts: saved image set and neighbor image set. A set of preferred images is called the saved image set, $Q = (q_1, q_2, ..., q_j)$, denotes all images that host peer has saved, which is used to refine the query for the purpose of learning host peer's current preference. The neighbor image set, $P_{ij} = (p_{i1}, p_{i2}, ..., p_{ik})$, denotes image *j* recommended by neighbor *i* which is used to calculate similarity of the neighbor *i*. The neighbor image set includes images a host peer has saved only.

Saved image set includes information about peer's preferences on images, while neighbor image set includes information about similarity of each neighbor according to their recommendations. The peer profile is constantly updated with newly obtained relevance images and similarity of its neighbors to dynamically reflect peer's most recent preference. This is significantly different from the user profiles used in traditional server based recommender systems [11].

3.2 Neighbor Definition

The agent of host peer *h* has a neighbor set N_h consists of peers who have recommended relevance images to host peer *h*. Each Neighbor has a $s(n_k, h)$ respectively. Here, $s(n_k, h)$ denotes the similarity between a neighbor n_k and host peer *h*. As equation (1), a similarity accumulates the reciprocal of the *Dist(p,Q)* calculated by CBIR agent is discussed in section *4.2* to reflect both the similarity and the success times of recommendations.

$$s_{i+1}(n_k, h) = s_i(n_k, h) + Dist(p, Q^h)^{-1} \qquad (1)$$

The neighbors of host peer *h* divided into two sets: the target peer set (N_h^t) and the candidate target peer set (N_h^c).

$$N_h = N_h^t \cup N_h^c \qquad (2)$$

Using the $s(n_k, h)$, the agent generates a set of target peers, $N_h^t = \{n_1, n_2, ..., n_k\}$, such that $s(n_1, h)$ is maximum, $s(n_2, h)$ is the next maximum, and so on where *k* is a limited number of target peers. The agent of a host peer forwards recommendations to its target peers only. Besides target peers, the rest of neighbors is included to candidate target peer set (N_h^c). But, neighbors are dynamically exchanged: adding new peers and moving between two sets to reflect host peer's current preference.

4. RECOMMENDER AGENT

The personal agent consists of two modules: CF agent, CBIR agent. In this section, we describe the roles of two agents respectively.

4.1 CF Agent

CF agent is implemented with a learning algorithm to model a host peer and with a set of neighbor peers. CF agent makes it possible to find more relevance neighbors. The agent updates the peer profile using feedback information (*saved or not*). The updated profile is used to choose whom to keep as neighbors and target peers but also refine the query. The refined query is passed to the CBIR agent to predict the relevance of images for current preference of the host peer.
CF agent can only generate recommendations of its host peer, and forward other peers' recommendations to its target peers.

Initialization To initialize a new peer, agents can use a different algorithm. This is implemented by initializing the peer with neighbors taken from an artificially created. To determine initial neighbor set for new host peer *p*, the agent calculates the similarity between new host peer *p* and other peers using all saved image set as follows:

The agent determine initial neighbors for a peer *p*, $B=\{b_1,b_2,...,b_m\}$ such that $p \notin B$ and $sim(p,b_1)$ is maximum, $sim(p,b_2)$ is the next maximum, and so on. Here, $sim(p,b)$ denotes the similarity between two peers *p* and *b*, and is calculated using equation [3][4].

$$Dist(P,Q)=\sqrt{\frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} dist^2(p_i,q_j)}{n_1 n_2}} \qquad (3)$$

$$dist(p_i,q_j)=\sqrt{\sum_{s=1}^{S}(p_{is}-q_{js})^2} \qquad (4)$$

Where n_1 and n_2 are the number of saved images of the peer *b* and peer *p* respectively, and p_i and q_j are the *i*th and *j*th image, and $dist(p_i,q_j)$ is a distance function between an image p_i and q_j. And in equation (4), *s* is the number of dimensions of the feature space and p_{is} and q_{js} are coordinates of an image p_i and q_j on the *s*th dimension respectively. It has characteristics of treating neighbors with similar saved images using the shortest distance.

The set of similar peers to new peer *p*, $B=\{b_1,b_2,...,b_m\}$, will be the agent's initial neighbor set. Additional neighbors will be found by the exchange of recommendations.

Recommendation Generation An agent generates recommendations to each other not on request but whenever a host peer saves a new image. However, each agent can only collaborate directly with its target peers, and hence to reach agent beyond the target peers they have to forward recommendations for each other. Figure 2 illustrates it.

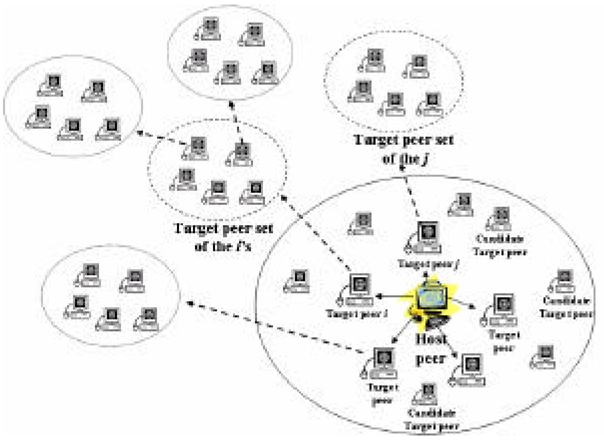

Figure 2. Communications of an agent with its target peer set

Neighbor Reformation The recommender agent dynamically selects which users to be included in neighbors. When a host peer receives a suitable recommendation, the recommending peer might be added to neighbor set. When an agent receives a forwarded recommendation from other peer ***i***, the recommending peer ***i*** might be added to the candidate target peer set and discards an old one ***j*** with the lowest similarity value as equation (5).

$$\begin{aligned}&if\ \ i \notin N\\ &then\ \ N_c \leftarrow N_c-\{j\}+\{i\}\end{aligned} \qquad (5)$$

Moreover, if a candidate target peer recommends relevant images continuously, it becomes a target peer. If the candidate target peer is more similar to the host peers than the target peer, the candidate target peer will replace the target peer and vice verse in each neighbor set.

$$\begin{aligned}&if\ \ arc\max_{n_{c_j}\in N_c}(sim(h,n_{c_j})) > arc\min_{n_{t_i}\in N_t}(sim(h,n_{t_i})\\ &then\ \ N_c \leftarrow N_c-\{n_{c_j}\}+\{n_{t_i}\}\ \&\ N_t \leftarrow N_t-\{n_{t_i}\}+\{n_{c_j}\}\end{aligned} \qquad (6)$$

According to this mechanism, an agent finds more similar neighbors resulting to be given relevant images. Thus an agent might keep better constituent neighbor members and recommendations should be reached to more similar users.

4.2 CBIR Agent

A recommender agent can receive from other recommender agent. All received images are put in a queue except the same as saved images. The CBIR agent uses all saved images of the host peer as multiple query points, and retrieves images based on distance between the multiple query points and images in the queue. For all images in the queue, this agent calculates the distances from the multiple query points and generates a list of *k* nearest images (*k-NN*s) as recommendations. This agent retrieves *k* images entirely based on visual features of query that represents the peer's current preference.

The calculation to retrieves *k* images is as follows: CBIR agent uses the all saved images as the query *Q*, and continuously refines the query by adding the query points in *Q* with the newly saved images. Since a query is allowed to have multiple query points, the distance function between an image *x* and a query *Q* should aggregate multiple distance components from each query points to the image. We use the following aggregate distance function:

$$Dist(x,Q) = \sqrt{\frac{g}{\sum_{j=1}^{g} 1/dist^2(x,q_j)}} \tag{7}$$

where *g* is the number of query points in a query *Q*, q_j is the *j*th query point of *Q*, and $dist(x,q_j)$ is a distance function between an image *x* and a query point q_j. We derived the Equation (7) from the FALCON's formula [14]. It has characteristics of treating an image with the shortest distance component to any one of query points as the image with the shortest aggregate distance.

Just as this system continuously refines the query in newly added images, it also updates the Equation (7) to reflect the peer's current preference. For this purpose, we define the $dist(x,q_j)$ in Equation (7) as:

$$dist(x,q_j) = \sqrt{\sum_{l=1}^{L} w_l (x_l - q_{jl})^2} \tag{8}$$

where *L* is the number of dimensions of feature space, w_l is a weight of the *l*th dimension in the feature space, and x_l and q_{jl} are coordinates of an image *x* and a query point q_j on the *l*th dimension, respectively. w_l in Equation (8) is substituted by $1/s_l$ at the end of every connection of the peer, where s_l is a standard deviation of coordinates of *l*th dimension of images. Note that s_l is calculated using all images in the relevant set accumulated. This distance function update is to better reflect a peer's current preference by allowing different weights by dimension and putting more emphasis on the features with smaller variance. Note that equal weights are to be used for all dimensions when there is only one query point in Q.

Based on the above definitions and discussions, CBIR agent generates Top *k* recommendation list for the host peer, $X=\{x_1, x_2 ... x_k\}$ such that $Dist(x_1, Q)$ is the minimum, $Dist(x_2, Q)$ is the next minimum, and so on.

The retrieved *k* images are presented when the peer connects the network. And the peer skims through the list to see if there are any images of interest. Then, the peer may save desired images on the peer's computer. This relevance judgment on the presented images passes to CF agent to learn the peer's current preference.

5. EVALUATION

In order to performance evaluation the proposed image recommender system we implement a simulator with variable parameters: size of the neighbor set, exploration propensity of the user for simulated 30 days. The components of the simulator are some parts of the recommender system. One of the simulated parts is images including their representation as vectors in a nine dimensional feature space for HSV values. For the simulations, the 1000 images that Korea Telecom Freetel (KTF), a leading Korean CDMA carrier, is currently offering were used. Other parts consist of 1000 peers modeled as having distinctive preference for some images. And they have initial neighbor set to exchange recommendations. Last, the recommender agents work as described in section 4.

Proposed recommender system is compared with centralized recommender system. In our system, each peer has a unique recommender agent with a limited neighbor set, while in the compared system, a single agent manages all users for recommendations. For performance evaluation of the system, we use metrics most widely used for recommender systems, *precision* and *recall* defined as follows:

Precision = recommended relevant images / all recommended images
Recall = recommended relevant images / total relevant images

The quality of recommendations is known to vary by the size of the neighbor set, the number of TTL. The metrics for each individual peer were computed and the average value for use as the metric was calculated. We watch how performance of the system changes for every simulation day.

For analysis of experiment results, two statistical tests are conducted. One is the *t* test for comparison of the average performance of our system with the centralized system. The other is the two-way ANOVA test with repetition for assessment of the effects of two factors, size of the neighbor set and *n*, on *ttl*.

6. CONCLUSION

We propose a recommender system based on a distributed P2P architecture that has originated with user-oriented principle rather than business itself. This could extend research area of the recommender systems to distributed architecture.

We develop an adequate solution for image recommendations in distributed P2P architecture, combines two techniques: collaborative filtering and content-based image retrieval. The system supports peers of the P2P system in finding a desired image by generating personalized recommendations of images. We implemented fully functioning personal recommender agent that automatically selects neighbors and recommends contents by learning user preference from user's content usage without requiring user's explicit ratings. We believe that the suggested P2P based recommender system should provide the users with more qualified recommendations, while it reduces the effort and time of users.

The proposed system has flexible algorithm to share any contents to can be analyzed. Therefore our future work includes varying contents, such as music and text, and implementing other techniques besides CBIR for image.

REFERENCES

[1] Balabanovic, M., Y. Shoham, "Fab: Content-based Collaborative Recommendation", *Communication of the ACM*, Vol. 40, No. 3, pp 66-72, 1997.

[2] Canny, J., "Collaborative filtering with privacy", *In Proceedings of the IEEE Symposium on Research in Security and Privacy*, pp 45—57, 2002.

[3] Cho, Y.H., J.K. Kim, S.H. Kim, "A Personalized Recommender System Based on Web Usage Mining and Decision Tree Induction", *Expert Systems with Applications*, Vol. 23, No. 3, pp 329-342, 2002.

[4] Damiani, E., S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks", *In Proc. of the 9th ACM Conference on Computer and Communications Security*, pp 17-21, November 2002.

[5] Kim, C.Y., J.K. Lee, Y.H. Cho, D.H. Kim, "VISCORS: a Visual Contents Recommender System on the Mobile Web", *IEEE Intelligent Systems, Special issue on Mining the Web for Actionable Knowledge*, Accepted for publication, 2004.

[6] Kim D.H., C.W. Chung, "Qcluster: Relevance Feedback Using Adaptive Clustering for Content-Based Image Retrieval", *Proc. ACM SIGMOD International Conference*, pp 599-610, 2003.

[7] Melville, P., R.J. Mooney, R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations", *Proc. the 18th National Conference on Artificial Intelligence*, pp187-192, 2002.

[8] Olsson, T., "Bootstrapping and Decentralizing Recommender Systems", *PhD thesis, Uppsala University, Uppsala, Sweden*, 2003.

[9] Peng, H., X. Bo, Y. Fan, S. Ruimin, "A scalable P2P recommender system based on distributed collaborative filtering", *Expert Systems with Applications*, Vol. 27, pp 203–210, 2004.

[10] Porkaew K., K. Chakrabarti, S. Mehrotra, "Query Refinement for Multimedia Similarity Retrieval in MARS", *Proc. 7th ACM Multimedia Conference*, pp 235-238, 1999.

[11] Prete, C.D., J.T. McArthur, R.L. Villars, I.L. Nathan Redmond, D. Reinsel, "Industry developments and models, Disruptive Innovation in Enterprise Computing: storage", *IDC*, 2003.

[12] Sarwar, B., G. Karypis, J.A. Konstan, J. Riedl, "Analysis of recommendation algorithms for e-commerce", *Proceedings of the ACM Conference on Electronic Commerce*, pp 158-167, 2000.

[13] Schafer, J.B., J.A. Konstan, J. Riedl, "E-commerce recommendation applications", *Data Mining and Knowledge Discovery,* Vol. 5, No. 1-2, pp 115-153, 2001.

[14] Wu, L., C. Faloutsos, K. Sycara, T. Payne, "FALCON: Feedback Adaptive Loop for Content-Based Retrieval", *Proc. 26th VLDB Conference*, pp 297-306, 2002.

[15] Zhou, X.S., T.S. Huang, "Relevance feedback for image retrieval: a comprehensive review," *ACM Multimedia Systems Journal*, Vol. 8, No. 6, pp 536-544, 2003.