PAPER

Folksonomical P2P File Sharing Networks Using Vectorized KANSEI Information as Search Tags

Kei OHNISHI[†a)], ***Member***, **Kaori YOSHIDA**[†b)], ***Nonmember***, ***and*** **Yuji OIE**[†c)], ***Fellow***

SUMMARY We present the concept of folksonomical peer-to-peer (P2P) file sharing networks that allow participants (peers) to freely assign structured search tags to files. These networks are similar to folksonomies in the present Web from the point of view that users assign search tags to information distributed over a network. As a concrete example, we consider an unstructured P2P network using vectorized Kansei (human sensitivity) information as structured search tags for file search. Vectorized Kansei information as search tags indicates what participants feel about their files and is assigned by the participant to each of their files. A search query also has the same form of search tags and indicates what participants want to feel about files that they will eventually obtain. A method that enables file search using vectorized Kansei information is the *Kansei query-forwarding method*, which probabilistically propagates a search query to peers that are likely to hold more files having search tags that are similar to the query. The similarity between the search query and the search tags is measured in terms of their dot product. The simulation experiments examine if the Kansei query-forwarding method can provide equal search performance for all peers in a network in which only the Kansei information and the tendency with respect to file collection are different among all of the peers. The simulation results show that the Kansei query forwarding method and a random-walk-based query forwarding method, for comparison, work effectively in different situations and are complementary. Furthermore, the Kansei query forwarding method is shown, through simulations, to be superior to or equal to the random-walk based one in terms of search speed.
key words: *P2P file sharing, folksonomy, query forwarding, Kansei, human*

1. Introduction

Recently, peer-to-peer (P2P) network models have attracted a great deal of attention. The concept of the P2P network model is completely different from that of a conventional client-server network model. While a conventional server-client network model explicitly distinguishes hosts providing services (servers) from hosts receiving services (clients), a P2P network model does not assign fixed roles to hosts. Hosts composing P2P networks, referred to as *peers*, can be both servers and clients and provide services for each other with direct connection between them, so that P2P networks could be used to facilitate autonomic and decentralized service management.

One of the applications of P2P networks that has attracted interest is a distributed storage system for file sharing. A distributed storage system for file sharing provides a large amount of storage by accumulating unused storage of hosts, enabling large amounts of data to be stored and shared without the need for a costly file server. According to [1], there are several forms in P2P networks for file sharing. We can roughly classify P2P networks for file sharing into two types. One type is P2P networks that have a mechanism to manage file locations in a network. The other type is P2P networks that do not have such a mechanism.

One of the mechanisms to manage file locations in P2P networks is a distributed hash table (referred to hereinafter as DHT) [2]–[5]. P2P networks using DHT instruct peers in a network as to which files they require and how peers are connected to one another. Therefore, these network are generally referred to as structured P2P networks. An exhaustive survey on structured P2P networks is provided in [1]. In addition, P2P networks that do not have a mechanism to manage file locations in a network, which are referred to as unstructured P2P networks, cannot provide peers with the locations of requested files. Therefore, a query-forwarding method is needed to find requested files. Most studies on unstructured P2P networks have examined query-forwarding methods [6], [7].

With respect to information search techniques, the simplest method is to search for objects by explicitly expressing target objects such as file names. However, in order to express the target objects precisely, we need to know what to search for in advance, and such a situation is rare. Therefore, searching for objects from pieces of information included in target objects has been considered. The present Web search techniques adopt such a strategy. Pieces of information included in target objects are not target objects themselves, but can be thought of as alternative forms pointing to target objects, and the alternative forms can be regarded as a sort of search tags of target objects. Furthermore, as a variation of pieces of information included in target objects, we can use features that are extracted by processing the target objects.

Focusing on ways to assign search tags to shared information, in conventional Web information retrieval systems, some authorities have been assigning search tags to shared information in fixed ways, which is an top-down approach. Meanwhile, users have recently been interested in information retrieval systems that allow them to assign search tags to shared information as their ways. Such a bottom-up information retrieval system is referred to as a *folksonomy* [8]. At this moment, several folksonomies are running and most of them allow the users to freely assign keywords to shared

Manuscript received April 28, 2009.
Manuscript revised July 9, 2009.
[†]The authors are with Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka-shi, 820–8502 Japan.
a) E-mail: ohnishi@cse.kyutech.ac.jp
b) E-mail: kaori@ai.kyutech.ac.jp
c) E-mail: oie@cse.kyutech.ac.jp
DOI: 10.1587/transinf.E92.D.2402

information [9], [10].

Besides keywords, one of the possible forms of search tags that people can assign to shared information is *Kansei* information, which in Japanese means 'human sensitivity' information. There have been several attempts to use *Kansei* information in a manner such that how people feel about objects is assigned as the search tags of the participant. Such information retrieval systems are referred to as *Kansei* information retrieval systems, and there exist several concrete *Kansei* information retrieval systems [11], [12]. However, existing *Kansei* information retrieval systems are built upon a client-server network model. In addition, the Kansei information retrieval systems are required to build a map between search objects and Kansei information as search tags in advance.

In the present paper, we propose the concept of file sharing P2P networks that allow the participants (peers) to freely assign structured search tags to files. Since we refer to the present Web information retrieval systems that allows users to freely assign search tags to shared information like web pages as *folksonomies*, we can refer to the P2P file sharing systems proposed in the present paper as *folksonomical* P2P file sharing systems.

As a concrete example of this concept, we consider an unstructured P2P network using vectorized Kansei information as structured search tags for file search. The vectorized Kansei information as search tags indicates what people feel about their files, and is assigned by the participant to each of their files. A search query also has the same form of the vectorized Kansei information and indicates what people want to feel about files that they will eventually obtain. One of the original ideas in this paper is that the Kansei information is transformed into search tags and queries by each participant in a P2P network. A mechanism that enables file search using the vectorized Kansei information is the *Kansei query-forwarding method*, which probabilistically propagates a search query from a peer making a query to peers that are likely to hold more files whose search tags is similar to the query. The similarity between the search query and the search tags is measured by their dot product.

The present paper is organized as follows. Section 2 briefly describes related work. We describe the motivation behind the present work and the basic concept of the proposed network in Sect. 3. In Sect. 4, we describe the proposed P2P file sharing network that uses Kansei information as search tags, as well as the Kansei query-forwarding method. In Sect. 5, we examine through simulations if the Kansei query-forwarding method can provide equal search performance to participants that are different from each other in Kansei information and the tendency of queries. In Sect. 6, search speed for each file search using the Kansei query-forwarding method is examined through simulations. Finally, Sect. 7 presents our conclusions.

2. Related Work

In this section, we describe studies related to Kansei information retrieval systems as well as P2P networks with query-forwarding methods using search tags, which are not flooding-based and random-walk-based methods. In addition, we mention researches on a fusion of folksonomines and distributed systems like P2P networks.

According to [13], in which a variety of P2P networks with query-forwarding methods using search tags (metadata) are introduced, there are two types of P2P networks that use search tags. One type of P2P network can provide deterministic routing for queries with a form of search tags by means of a mechanism to locate files without question to which search tags is assigned, such as DHT. The other type of P2P network employs non-deterministic routing due to the lack of a mechanism to locate files without question. Most of the P2P networks introduced in [13] use text data as search tags.

The P2P network presented herein does not have a mechanism to manage file locations in the network. The reason for choosing this form of P2P network is as follows. Since Kansei information should vary among individuals, a map between files and Kansei information as search tags for the files could be a one-to-many map. In this case, it is difficult to generate a framework of P2P networks that is equipped with deterministic routing, such as DHT-based P2P networks, in which the basic assumption is that a map between files and search tags for the files is a many-to-one map. The P2P network proposed herein uses individual Kansei information (impression) with a form of a numerical vector having a fixed length as search tags for the files. Such P2P networks have not yet been investigated.

File search in the proposed P2P network is conducted using a query forwarding method that propagates a query in the form of vectorized Kansei information over a network, which is referred to herein as *the Kansei query forwarding method*. The Kansei query forwarding method basically forwards a query to peers that have more files with search tags similar to the query. In this case, a map between files and their search tags is a one-to-many map. Meanwhile, for the case in which a map between files and their search tags is a many-to-one map, such a query forwarding method is referred to as semantic routing [13]–[15]. The focus of semantic routing is literally semantics, and semantic routing usually employs text data extracted from text files as their search tags.

Besides search tags that participants freely assign to files in folksonomical P2P networks, reputation and recommendation is information that includes users' judgement or description in P2P networks. P2P networks utilizing reputation and/or recommendation have been well studied [16]–[19]. In both of P2P networks utilizing reputation and/or recommendation and folksonomical P2P ones, it is common that users are explicitly or implicitly asked to judge or describe objects in P2P networks such as persons, files, nodes, and so on.

However, reputation and recommendation is not search tags. In fact, to give recommendation or reputation in P2P networks, we need another mechanism to make recommendation or reputation and input information for that mechanism. For instance, in Tribler proposed in [17], the Recommendation Engine that is common to all peers analyzes preference of each peer and then recommends information to the peer based on the analysis. Peers can request to make recommendation or reputation to the P2P network. However, this request is not a search tag itself but just a trigger to have recommendation or reputation.

Meanwhile, in the folksonomical P2P file sharing network presented in this paper, the participants just search files for the network using search tags and then retrieve the files. Therefore, we think that a P2P network utilizing reputation and/or recommendation and the folksonomical P2P one can be integrated together and also that they are not compareable opponents. Unlike normal P2P networks, the folksonomical P2P network allows the participants to freely assign search tags to files. In this paper, we adopt vectorized Kansei information as search tags that participants can freely assign to files. The reason for allowing the participants to freely assign search tags to files, that for choosing vectorized Kansei information as the search tags, and the expected effects of using Kansei information will be described in Sect. 3 in details.

Next, existing Kansei information retrieval systems [11], [12], [20] have a common approach to information retrieval. First, these systems are built on server-client networks. Second, these methods are composed of two spaces and a map between them. One of the two spaces is of search objects, such as multi-media content. More precisely, one of the two spaces is of features extracted from the objects. The other space is of the impressions that participants form regarding the search objects. The fundamental difference among the existing systems is in a way to build a map between the two spaces mentioned above. On the other hand, the P2P network presented in this paper does not retrieve information by means of such a map between the two spaces.

3. Motivation and Concept

Although the goal of conventional information retrieval is usually to find desired information, the motivation of the present study is not to develop more efficient search techniques. Rather, we focus herein on information as intermediates that facilitate communication between humans. Although conventional information retrieval techniques recognize the identity of information itself, they do not assume the identity of the participant holding the information. Since there may be personal reasons for holding specific information, if we can understand these reasons during information retrieval, we may improve inter-participant communications beyond simple information retrieval.

The proposed approach, considering the identities of information holders in information retrieval systems, allows participants to assign search tags to their files as they see fit. In this case, different search tags can be assigned to identical information by different participants. This situation is not desirable in terms of efficient information retrieval. However, once a participant encounters other participants that assign the same or similar search tags to identical information through information retrieval, it is expected that the participants will share a great deal of information.

Next, we have to consider the form of search tags that participants are allowed to assign to their information in order to facilitate communication between the participants. An extreme method is to allow participants to assign search tags in a completely free manner. However, this would not realize practical information retrieval because the size of the search space can become large. Therefore, in the present study, we consider structured search tags, such as vectorized information, and the number of structured search tags types that participants can assign to their information is finite. In the present paper, we employ vectorized Kansei information as structured search tags. We will mention the reason for choosing vectorized Kansei information in the beginning of Sect. 4.

Finally, we must examine whether search tags that a certain search technique uses for information retrieval is useful for improving communication between participants, in which case the first information retrieval process is simply a trigger. However, this is difficult to examine because a model of human communications beyond actions of information retrieval is needed. Therefore, the simulation experiments in Sect. 5 examine whether such search tags helps to provide opportunities for a participant to meet other participants with a similar sense regarding their files.

4. Kansei Information in P2P Networks

In the present paper, we assume P2P networks for file sharing that do not have a mechanism to manage file locations. Under the condition in which there is no constraint on files that peers are allowed to hold, we can expect that files that participants (peers) in a network hold reflect the Kansei information of the participants. For instance, different participants may like different music or movies, based on personal preferences. If participants in a network assign information on how they feel about the contents of their files to these files and then release the Kansei information of these files to each other, the participants can not only learn what files are owned by other participants, but they may also learn how other participants feel about these files. This learning process could help the participant to communicate with each other beyond actions of information retrieval. That is why we choose vectorized Kansei information as structured search tags.

4.1 Vectorized Kansei Information as Search Tags

The simplest conventional file search mechanism requires explicit information to be given regarding search objects, such as file names or pieces of file names. Here, we consider file search with search tags as a query, such that file search begins by specifying a name or feature of a set to which search objects belong and eventually reaches concrete desired objects. A name or feature of a set to which files belong is upper concept than a file itself.

A set to which files belong consists of two sub-sets. One sub-set is related to a category, which is objective, such as the "music" category. The other sub-set is related to the impression or feeling of participants regarding the contents of files, which is quite subjective. Some participants may consider certain music to be pleasant, while other participants consider the same music to be unpleasant. In this case, a "music" file may belong to different subsets of impression or feeling, depending on the participant.

Next, one method of representing which sets a file belongs to is to use words such as "music" and "pleasant". Here, we use not words, but vectors of numerical values of fixed length, which enables the calculation of the dot product mentioned later herein. Owners of files are not allowed to completely freely assign vectors of numerical values as search tags to their files. Vectors of numerical values that owners of files are allowed to assign to their own files are determined in advance, and participants can assign a vector of numerical value to each file.

For example, a file that belongs to the "music" category is represented by (text, music, movie)= $(0, 1, 0)$, where "1" indicates that the file belongs to the category represented by the category name, and "0" indicates that the file does not belong to the category. Meanwhile, with respect to a numerical vector representing impression or feeling, a file is, for example, represented by (light, pleasant, beautiful)= $(-1, 0, 1)$, where "1" indicates that the content of the file gives the impression represented by the impression word, "0" indicates that the content of the file gives the impression represented by the impression word to an extent, and "-1" indicates that the content of the file does not give the impression represented by the impression word at all.

Participants may wish to add elements of a numerical vector to represent files during file sharing. For example, "complicated" is added into a numerical vector representing impression or feeling as its element. In a situation in which the elements of a numerical vector representing a file increase dynamically, the participants need to guarantee that consistent numerical vectors are assigned to their files. One solution for this is to introduce a server that manages elements of a numerical vector to represent files. The server accepts the registration of elements from participants and then forms a numerical vector by arranging the elements from the left side in old order. Participants obtain information on a new numerical vector regularly. In this way, even if a participant does not have information on a new numerical vector, there is no inconsistency in terms of the order of elements between a numerical vector that the participant knows and a new numerical vector up to the length of the numerical vector that the participant knows.

In addition, it may not be practical to force participants to assign a numerical vector to all of their files with the increasing number of files. One solution for this is to assign a certain numerical vector, for example (text, music, movie, light, pleasant, beautiful) = $(0, 0, 0, 0, 0, 0)$, to all files as a default or to simply ignore the elements of numerical vectors that are not filled in by participants when performing operations on numerical vectors, as explained later herein.

While each file is represented by a numerical vector as mentioned above, a peer that would hold several files is represented by the sum of all numerical vectors for the files. The representation of a peer is used for query forwarding mentioned later. For example, when a peer holds three files with vectors as search tags, the representation of the peer is the sum of the three vectors. In addition, it is possible to make participants select files for the representation of peers.

A search query made by a peer is represented in the form of a numerical vector assigned to a file as search tags, such as (text, music, movie, light, pleasant, beautiful) = $(1, 0, -1, 1, 1, 0)$. A search query is, as explained in the following section, propagated with high probability to peers in which the representations give larger values of the dot product with the query. The number of hops that is allowed for a query is limited to N_h, and lists of files of peers to which the query was propagated are given to a peer making the query. The peer making the query can select and download some of the files in these lists, if the peer desires so.

4.2 Kansei Query Forwarding

The proposed P2P network for file sharing has a mechanism that varies the probability with which a query is propagated on network links through file searches. The probability assigned to network links is not managed by particular peers, but varies in a self-organizing manner through a file search performed by each peer. In the following, we will explain the Kansei query-forwarding method, which plays a central role in self-organization as well as in file search.

Here, we let the number of walkers propagating a query be one, although more than one walker can be used. The walker considers a numerical vector $\boldsymbol{q}$, such as (1, 0, -1, 1, 1, 0), to be a query. The walker obtains values of the dot product between $\boldsymbol{q}$ and N_n numerical vectors representing the peers adjacent to the peer at which the walker is currently located, $\boldsymbol{r}_i$ $(i = 1, 2, \cdots, N_n)$. The probability with which the k-th peer in the N_n peers is selected as the peer that the walker will hop, ps_k, is denoted by Eq. (1).

$$ps_k = \frac{dp_k - min_{dp} + 1}{\sum_{i=1}^{N_n} \{dp_i - min_{dp} + 1\}}, \tag{1}$$

where $dp_k = \boldsymbol{q} \cdot \boldsymbol{r}_k$, and min_{dp} is the smallest value among the N_n dot products. Since the dot product can be negative, all of the dot products are forced to be positive by means of the term $-min_{dp} + 1$. In doing so, the peer to which the walker will hop next is determined with a probability proportional to the dot product between its numerical vector and the query. However, if the peers that a walker has visited once in the present file search are again selected, according to Eq. (1), the walker selects a peer that has not been visited

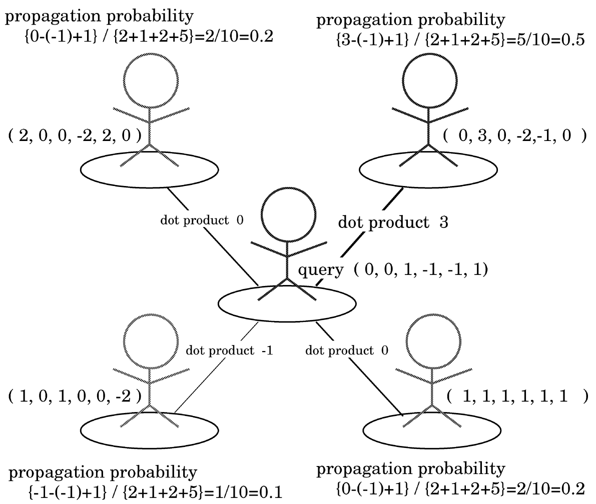


Fig. 1 Example of query forwarding. Circles represent peers, and a query has reached the peer in the center. The representations of the peers linked to the peer in the center are described near the circles. Calculating the values of the dot product between the query using a form of a numerical vector and the numerical vectors representing the peers, the probability of the query is propagated to each peer, as given by Eq. (1).

by the walker for its next hop. An example in which the proposed Kansei query-forwarding method decides a peer as the next hop of a walker is shown in Fig. 1.

Lists of files of peers that a walker visited within a limited number of hops, N_h, are provided for a peer making a query. Then, if the peer making the query finds files that it wants in the lists, the peer can download files within a limited number of downloads, as mentioned in the next section. In addition, the peer making the query also obtains the numerical vectors representing the files that it downloaded. When a peer making a query downloads several files from peers that a walker visited in the present file search, files held by the peer making the query vary. Consequently, the representation of the peer, which is the sum of all of the numerical vectors attached to its own files, also varies. The peer making the query can also modify the numerical vectors of the files that it has downloaded. Therefore, the probability with which a query is propagated on links to the peer making the query varies.

4.3 P2P File Sharing

The proposed P2P network has a restriction on the number of downloads to peers in the network, such that peers are not allowed to download an unlimited number of files from other peers, but rather are allowed to download files according to the number of times that other peers have downloaded their files. When a peer has had one of its own files downloaded by a peer, the peer increases by one the number of files that it can download from other peers. On the other hand, when a peer has downloaded one file from another peer, it decreases by one the number of files that it can download from other peers. Peers that hold popular files are likely to increase the number of times that they can download files from others. However, every peer is given a fixed number of times that it can download files, N_D, when it participates in the network, because without doing so, the P2P network does not work as a system for file sharing. After performing N_D complete downloads, peers can only increase the number of downloads by having their files downloaded by other peers.

As explained in the previous section, numerical vectors representing both query and file are given by peers, and changes in the numerical vectors cause changes both in the probability with which a query is propagated to the peers and in the search results. Therefore, acts of giving numerical vectors to files are strategic factors for the purpose of file sharing. However, even if queries have frequently reached certain peers, the peers can not increase the number of downloads unless their files are actually downloaded by other peers.

5. Experimental Evaluation

5.1 Objective

The P2P file search considered in the present paper relies on Kansei information of participants in the network. Kansei information is generally different for each participant, so that the search performance for each participant might not be the same. However, from the viewpoint of P2P file sharing, in which all of the participants (peers) are equal in terms of function, it is not good for certain participants to take advantage of P2P file search due to their Kansei information. Therefore, we experimentally examine whether the Kansei query-forwarding method can provide equal search performance for the situation in which the Kansei information of participants in the network is diverse. Therefore, we need to model differences among peers in Kansei information, as well as differences among peers with respect to query tendencies. We will describe the modeling of these differences later herein.

The Kansei information of participants which is produced by the model presented here might not be similar to that in the real world. However, as explained later, the model has a parameter to change size of difference among the Kansei information of participants, and we can produce a variety of the Kansei information of participants by adjusting the parameter value of the model.

5.2 P2P Simulation Model

With respect to network structure, the number of peers, n_p, present in the network is 200. The network topology used is full-mesh. The reason for choosing a full-mesh topology is that in the present paper we intend to have a network topology including a set of the propagation probability that is stabilized through repeating file searches, not by having peers choose their links. However, a full-mesh topology might be impractical with an increasing number of peers, and, in practice, it may be necessary to have all peers select a fixed number of links. Next, the number of files types distributed over the network, f_k, is the same as the number of peers, that is 200. Initially, each peer has a different file.

All of peers conduct query generation in turn. A peer whose turn has arisen generates a query only once. However, if a peer whose turn has arisen dose not have permission of downloading files, then the peer is just skipped.

5.3 Model of Kansei Information

We use one-dimensional and two-dimensional numerical vectors as vectorized Kansei information assigned to files, and in this section, we conduct evaluation experiments for both cases of one-dimensional and two-dimensional vectorized Kansei information. In both cases, an element of vector takes the value of either "-1" or "$+1$" and does not take the value of 0. The Kansei information of each peer (participant) in the network is represented as the value (-1 or $+1$) that each of the n_p peers gives to the f_k types of files that are distributed over the network.

Although peers cannot assign impressions or feelings to files prior to seeing the files, we define which values each peer assigns to f_k types of files in advance. Therefore, the Kansei information of each peer for f_k types of files is represented as a numerical vector with f_k elements in the case of one-dimensional vectorized Kansei information, as $(-1, -1, \cdots, +1)$, in which the i-th element corresponds to the i-th file type among f_k file types, and is represented as a numerical vector with $2f_k$ elements in the case of two-dimensional vectorized Kansei information. We hereinafter refer to this numerical vector as the *Kansei vector*.

The Kansei vector of each peer in the simulation provides impression or feeling not only about files that the peer actually holds but also about files that the peer has not met yet. However, this does not mean that the representation of each peer is produced using the Kansei information for files that the peer does not have. The representation of each peer at a certain moment is always produced only from the Kansei information for files that the peer actually holds. Therefore, the Kansei query forwarding is also executed based on the representations of peers that are produced from the Kansei information for files that they actually have. In addition, the Kansei vector of each peer is, as explained in Sect. 5.4, used for determining query tendency of the peer. We adopt this as one of methods for determining query tendencies of peers.

The Kansei vector of each peer is generated by modifying a given numerical vector with f_k elements in the case of one-dimensional vectorized Kansei information and with $2f_k$ elements in the case of two-dimensional vectorized Kansei information. The given numerical vector without modification is referred to hereinafter as the *prototypical Kansei vector*. The prototypical Kansei vector is modified by flipping a value of each element (-1 to $+1$ or $+1$ to -1) with probability p_b. The expected Hamming distance between two Kansei vectors, H_A, is generated according to following equation.

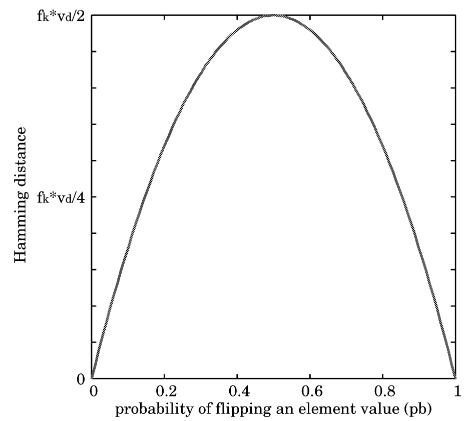


Fig. 2 The expected Hamming distance between two generated Kansei vectors, in which f_k is the number of file types distributed over the network and v_d is the number of elements of vectorized Kansei information assigned to each of file type.

$$H_A = f_k \times v_d \times \{p_b \times (1 - p_b) + (1 - p_b) \times p_b\}$$
$$= 2f_k \times v_d \times p_b(1 - p_b), \qquad (2)$$

where f_k is the number of file types distributed over the network and v_d is the number of elements of vectorized Kansei information assigned to each of file type. When the expected Hamming distance is small, that is when the value of p_b is small, vectorized Kansei information works effectively as search tags for the Kansei query forwarding method. In fact, we will show in Sect. 6 that when the expected Hamming distance is smal, the Kansei query forwarding method is better than the one-walker random walk for comparison in terms of search speed. In addition, the expected Hamming distance is shown in Fig. 2.

Concretely, we use the three types of prototypical Kansei vectors below to produce the Kansei vector of each peer for both cases of one-dimensional and two-dimensional vectorized Kansei information.

Case of one-dimensional vectorized Kansei information

The three prototypical Kansei vectors below differ from each other only in the ratio of "-1" to "$+1$".

Prototypical Kansei vector (1):

The 1st through 100th elements in the vector take the value of -1, and the other elements take the value of $+1$. Figure 3 shows how to produce the Kansei vector of each peer in the network from the prototypical Kansei vector (1).

Prototypical Kansei vector (2):

The 1st through 120th elements in the vector take the value of -1, and the other elements take the value of $+1$.

Prototypical Kansei vector (3):

The 1st through 160th elements in the vector take the value of -1, and the other elements take the value of $+1$.

Figure 4 shows the average numbers of "-1" values in the Kansei vectors of 200 peers produced using prototypical Kansei vectors (1), (2), and (3) and different values of p_b. Figure 4 shows that the average number of "-1" values becomes approximately 100 at $p_b = 0.5$ independent of which

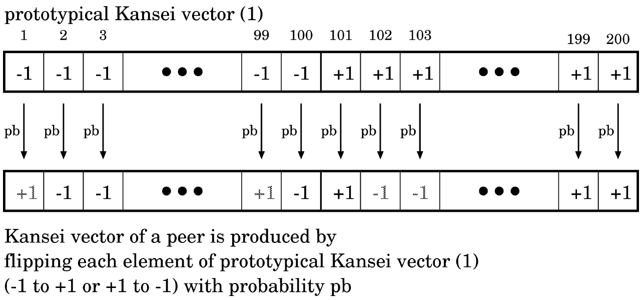


Fig. 3 Method by which to produce the Kansei vector of each peer in the network from the prototypical Kansei vector (1).

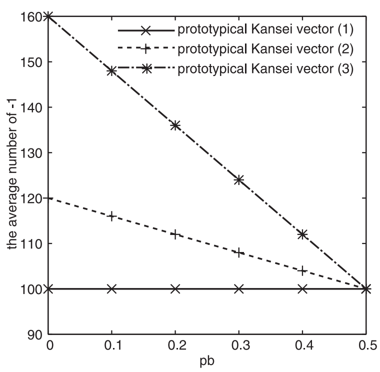


Fig. 4 Average numbers of "−1" values in the Kansei vectors of 200 peers produced using prototypical Kansei vectors (1), (2), and (3) and different values of p_b.

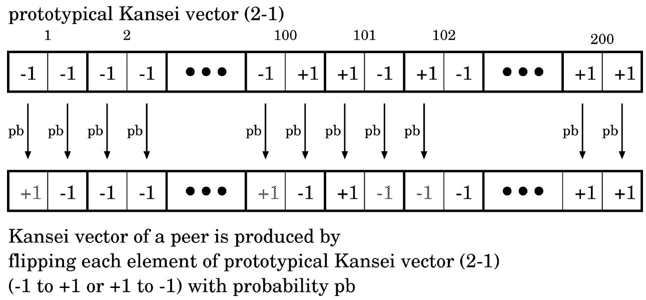


Fig. 5 Method by which to produce the Kansei vector of each peer in the network from the prototypical Kansei vector (2-1).

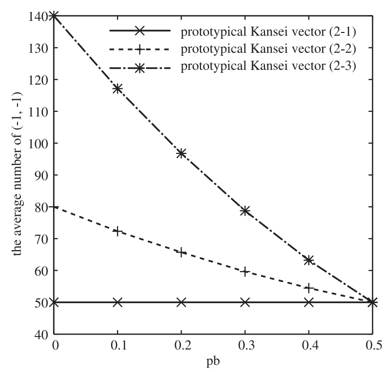


Fig. 6 Average numbers of $(-1,-1)$ in the Kansei vectors of 200 peers produced using prototypical Kansei vectors (2-1), (2-2), and (2-3) and different values of p_b.

prototypical Kansei vector is used.

Case of two-dimensional vectorized Kansei information

The three prototypical Kansei vectors below differ from each other in the ratio of $(-1,-1)$ to $(+1,+1)$", but have the same ration of $(-1,+1)$ to $(+1,+1)$.

Prototypical Kansei vector (2-1):

The 1st through 50th files take the value of $(-1,-1)$, the 51st through 100th files take the value of $(-1,+1)$, the 101st through 150th files take the value of $(+1,-1)$, and the 151st through 200th files take the value of $(+1,+1)$. Figure 5 shows how to produce the Kansei vector of each peer in the network from the prototypical Kansei vector (2-1).

Prototypical Kansei vector (2-2):

The 1st through 80th files take the value of $(-1,-1)$, the 81st through 120th files take the value of $(-1,+1)$, the 121st through 160th files take the value of $(+1,-1)$, and the 161st through 200th files take the value of $(+1,+1)$.

Prototypical Kansei vector (2-3):

The 1st through 140th files take the value of $(-1,-1)$, the 141st through 160th files take the value of $(-1,+1)$, the 161st through 180th files take the value of $(+1,-1)$, and the 181st through 200th files take the value of $(+1,+1)$.

Figure 6 shows the average numbers of $(-1,-1)$ in the Kansei vectors of 200 peers produced using prototypical Kansei vectors (2-1), (2-2), and (2-3) and different values of p_b. Figure 6 shows that the average number of $(-1,-1)$ becomes approximately 50 at $p_b = 0.5$ independent of which prototypical Kansei vector is used.

5.4 Model of Tendency of Queries

A search query is of the same form as Kansei information as search tags, and as a value of either "−1" or "+1" in the case of one-dimensional vectorized Kansei information and as either of $(-1,-1)$, $(-1,+1)$, $(+1,-1)$, and $(+1,+1)$ in the case of two-dimensional vectorized Kansei information.

In the case of one-dimensional vectorized Kansei information, the tendency of a peer with respect to queries is expressed by the probability with which "−1" is chosen as a query by the peer. This probability is denoted by $p \in [0,1]$, where the probability with which "+1" is chosen is $1-p$.

In the case of two-dimensional vectorized Kansei information, the tendency of a peer with respect to queries is expressed by the probability with which $(-1,-1)$, $(-1,+1)$, $(+1,-1)$, and $(+1,+1)$ are chosen as a query by the peer. This probability is denoted by $p1 \in [0,1]$ for $(-1,-1)$, $p2 \in [0,1]$ for $(-1,+1)$, $p3 \in [0,1]$ for $(+1,-1)$, and $p4 \in [0,1]$ for $(+1,+1)$, where $p1+p2+p3+p4 = 1$.

We use two methods to determine the value of p and the values of $p1$, $p2$, $p3$, $p4$ for each peer in the network below.

Case of One-dimensional Vectorized Kansei Information

Random determination of p

This method literally determines p of each peer by a uniform random number in $[0, 1]$.

Kansei-correlated determination of p

This method determines p of each peer using the following equation:

$$p = N_0/f_k, \tag{3}$$

where N_0 is the number of -1 values in the Kansei vector of the peer, f_k is the number of file types distributed over the network, and f_k is 200.

Case of Two-dimensional Vectorized Kansei Information

Random determination of $(p1, p2, p3, p4)$

This method determines $(p1, p2, p3, p4)$ of each peer using the following equations:

$$p1 = p^*1/(p^*1 + p^*2 + p^*3 + p^*4), \tag{4}$$
$$p2 = p^*2/(p^*1 + p^*2 + p^*3 + p^*4), \tag{5}$$
$$p3 = p^*3/(p^*1 + p^*2 + p^*3 + p^*4), \tag{6}$$
$$p4 = p^*4/(p^*1 + p^*2 + p^*3 + p^*4), \tag{7}$$

where p^*1, p^*2, p^*3, and p^*4 are a uniform random number in $\left[10^{-6}, 1\right]$.

Kansei-correlated determination of $(p1, p2, p3, p4)$

This method determines $(p1, p2, p3, p4)$ of each peer using the following equation:

$$p1 = N_1/f_k, \tag{8}$$
$$p2 = N_2/f_k, \tag{9}$$
$$p3 = N_3/f_k, \tag{10}$$
$$p4 = N_4/f_k, \tag{11}$$

where N_1, N_2, N_3, and N_4 are the number of $(-1, -1)$, $(-1, +1)$, $(+1, -1)$, and $(-1, -1)$ in the Kansei vector of the peer, respectively, f_k is the number of file types distributed over the network, and f_k is 200.

5.5 File Search and Sharing

The number of walkers that propagate a query is just one. The number of hops that a walker is allowed during one file search is three, that is, E_h is 3. Each peer is initially given permission to download five files, that is, N_D is 5.

For example, when a peer makes a query of "-1", and then some of the peers that a walker visited have files that belong to the set of "-1" for the peer making the query and the peer making the query does not have, the peer chooses one of these files randomly and downloads the file. If the walker could not find a file that the peer does not have, the file search is finished.

5.6 Evaluation Criteria

Query forwarding methods are evaluated according to the number of file searches that are required so that all of the peers will hold more than $2n_p/5$ files, where n_p is the total number of file types distributed over the network and n_p is 200. This number of file searches is referred to as the *convergence time* and the state in which all of the peers hold $2n_p/5$ or more files is defined as the *convergence*. A failure of convergence indicates that convergence does not occur within the number of file searches given in advance, E_s. We set the value of E_s to 30,000. Quick convergence means that all of the peers in the network obtain similar and good search performance.

We count the following three cases as one file search: (1) a file that a peer making a query desires was found within three hops ($N_h = 3$) of a walker, (2) a file that a peer making a query desires was not found within three hops of a walker, and (3) a peer that did not have permission to download files from others when its turn arose. We do not count as one file search the situation in which some peer made a query but previously held all files belonging to the set represented by the query.

5.7 Simulation Settings

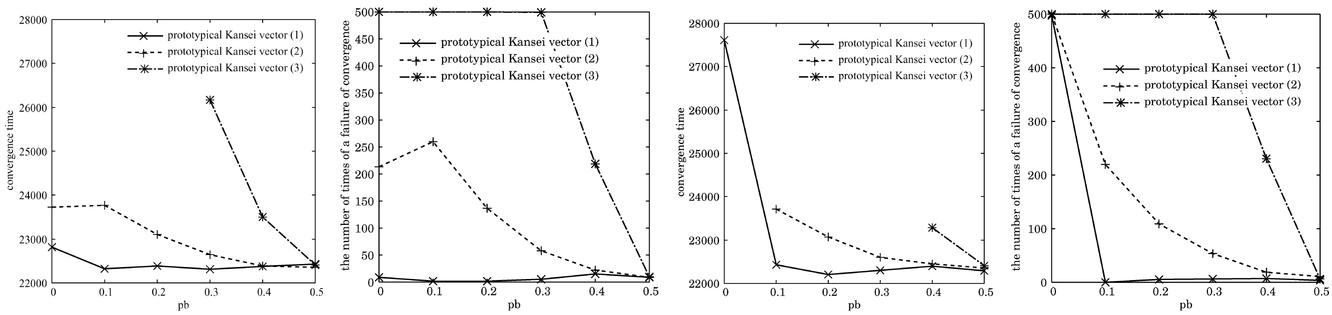
As mentioned earlier, three types of prototypical Kansei vectors are used, prototypical Kansei vectors (1), (2), and (3) in the case of one-dimensional vectorized Kansei information, and prototypical Kansei vectors (2-1), (2-2), and (2-3) in the case of two-dimensional vectorized Kansei information. These vectors are modified into the Kansei vectors of peers by flipping the values of each of their elements with probability p_b. In addition, two methods by which to determine the p values of peers in the case of one-dimensional vectorized Kansei information and the values of $(p1, p2, p3, p4)$ of peers in the case of two-dimensional vectorized Kansei information are used.

We will test all six combinations of three types of prototypical Kansei vectors and the two methods for determining probability related to query tendencies. For each combination with a value of p_b less than 0.5, we will observe the average convergence time over independent runs that resulted in convergence and the number of convergence failures during 500 independent runs.

For comparison, as a query forwarding method, we employ the *one-walker random walk*. The one-walker random walk allows a walker that knows a search query to randomly decide the next peer to visit. However, in one file search, a walker does not revisit the peers that it has already visited.

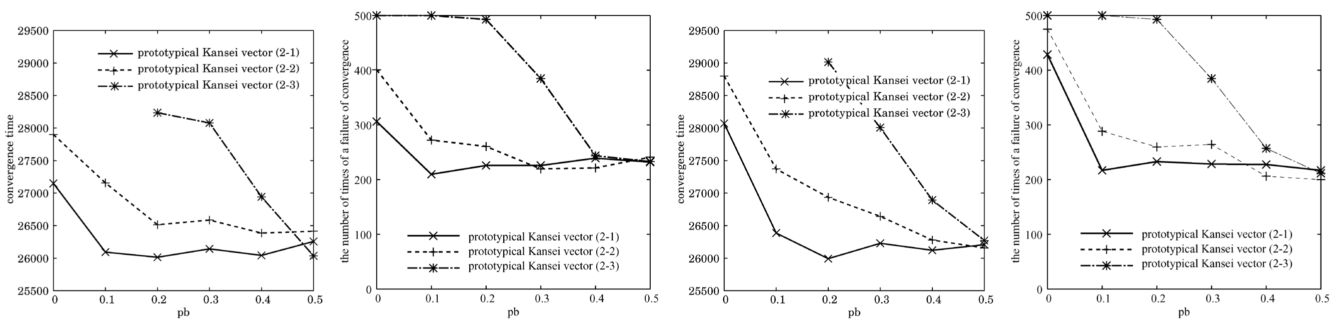
5.8 Results

The simulation results for the case of one-dimensional vectorized Kansei information are shown in Figs. 7 and 9. The



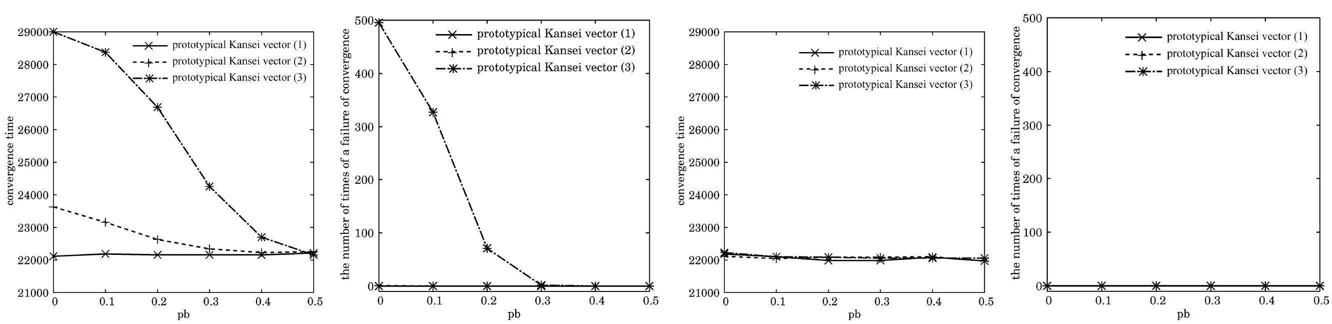
(a) Average convergence time. (b) Number of failures to converge. (c) Average convergence time. (d) Number of failures to converge.

Fig. 7 Simulation results obtained using random determination of p in the case of one-dimensional vectorized Kansei information. Sub-figures (a) and (b) are for the Kansei query-forwarding method, and sub-figures (c) and (d) are for a one-walker random walk.



(a) Average convergence time. (b) Number of failures to converge. (c) Average convergence time. (d) Number of failures to converge.

Fig. 8 Simulation results obtained using random determination of $(p1, p2, p3, p4)$ in the case of two-dimensional vectorized Kansei information. Sub-figures (a) and (b) are for the Kansei query-forwarding method, and sub-figures (c) and (d) are for a one-walker random walk.



(a) Average convergence time. (b) Number of failures to converge. (c) Average convergence time. (d) Number of failures to converge.

Fig. 9 Simulation results obtained using the Kansei-correlated determination of p in the case of one-dimensional vectorized Kansei information. Sub-figures (a) and (b) are for the Kansei query-forwarding method, and sub-figures (c) and (d) are for a one-walker random walk.

simulation results for the case of two-dimensional vectorized Kansei information are shown in Figs. 8 and 10.

Figures 7 shows the average convergence time and the number of convergence failures for the Kansei query-forwarding method and the one-walker random walk in the case of random determination of p, and Fig. 9 shows the average convergence time and the number of convergence failures for the Kansei query-forwarding method and the one-walker random walk in the case of Kansei-correlated determination of p.

Figures 8 shows the average convergence time and the number of convergence failures for the Kansei query-forwarding method and the one-walker random walk in the case of random determination of $(p1, p2, p3, p4)$, and Fig. 10 shows the average convergence time and the number of convergence failures for the Kansei query-forwarding method and the one-walker random walk in the case of Kansei-correlated determination of $(p1, p2, p3, p4)$.

The obtained results do not show the success rate in one file search by the used query forwarding methods, but rather indicate the success rate of the convergence defined in Sect. 5.6. The success convergence could result from smooth circulation of the permission to perform file downloads, which initially is given to each peer. Meanwhile, failure to converge may be a result of the permission to download being biased to specific peers or may be a result of the query forwarding method used being unable to provide reliable search performance for most peers.

Finally, every peer has few files just after the simulation starts. In this special situation, we might be able to observe specific behaviors of the Kansei query forwarding method and the one-walker random walk. However, in this paper, we focus on long-term observation as the convergence time.

5.9 Discussion

5.9.1 Case of Random Determination of p

This is the case of one-dimensional vectorized Kansei information.

According to Fig. 7 (comparing Figs. 7 (a) and 7 (b) for Kansei query forwarding and Figs. 7 (c) and 7 (d) for one-walker random walk), when using the random determination of p, the Kansei query forwarding method is superior to or approximately equal to the one-walker random walk in terms of the ability to induce quick convergence, that is, in providing similar search performance for all of the peers. Specifically, when using small values of p_b, the Kansei query forwarding method is superior to the one-walker random walk, and the ability of these two methods becomes more similar with increasing values of p_b.

When the difference in the ability to induce convergence between the Kansei query forwarding method and the one-walker random walk was the largest, that is, when using the prototypical Kansei vector (1) and $p_b = 0$, vectorized Kansei information as search tags works efficiently as search tags, and the randomness in file search provides redundancy. In addition, since the number of "-1" and "$+1$" values in the Kansei vectors of the peers are the same and the p values of the peers are randomly determined, the total number of files corresponding to "-1" and that corresponding to "$+1$" in the network are approximately the same at any time during file search. Therefore, all of the peers with their p values had approximately equal abilities to obtain and provide files, independent of p value.

When using Kansei vector (1) and the larger values of p_b, vectorized Kansei information as search tags no longer works as efficient search tags, and randomness is needed in the file search. However, unreliable search tags cause file search by the Kansei query forwarding method to be similar to random search. As a result, with increasing values of p_b, the abilities of the Kansei query forwarding method and the one-walker random walk to induce convergence become increasingly similar.

Next, when using prototypical Kansei vector (2), as in the case of using prototypical Kansei vector (1), vectorized Kansei information as search tags is still reliable if the value of p_b is small. Therefore, the ability of the Kansei query forwarding method to induce convergence is better than that of the one-walker random walk. However, the number of "-1" values in the Kansei vectors of most peers should be larger than the number of "$+1$" values. Since the p values of the peers were randomly determined in this situation, peers that have a high probability to produce a query of "-1" could easily obtain and frequently provide files corresponding to "-1". Meanwhile, peers that have high probability to produce a query of "$+1$" could not easily obtain and frequently provide files corresponding to "$+1$". This should result in a difference in search performance between the peers, and consequently, the ability of the Kansei query forwarding method to induce convergence in the case of using prototypical Kansei vector (2) and small values of p_b would be worse than that in the case of using prototypical Kansei vector (1) and small values of p_b.

Furthermore, in the case of using prototypical Kansei vector (3) and small values of p_b, it should be more difficult for peers that have a high probability to produce a query of "$+1$" to easily obtain and frequently provide files corresponding to "$+1$", compared to the case of using prototypical Kansei vector (2) and small values of p_b. However, even when prototypical Kansei vector (2) or (3) is used, if the value of p_b becomes larger, the numbers of "-1" and "$+1$" values in the Kansei vectors of the peers become more similar and randomness is needed in the file search. Therefore, the abilities of the Kansei query forwarding method and the one-walker random walk to induce convergence would become closer with the increasing value of p_b.

5.9.2 Case of Random Determination of $(p1, p2, p3, p4)$

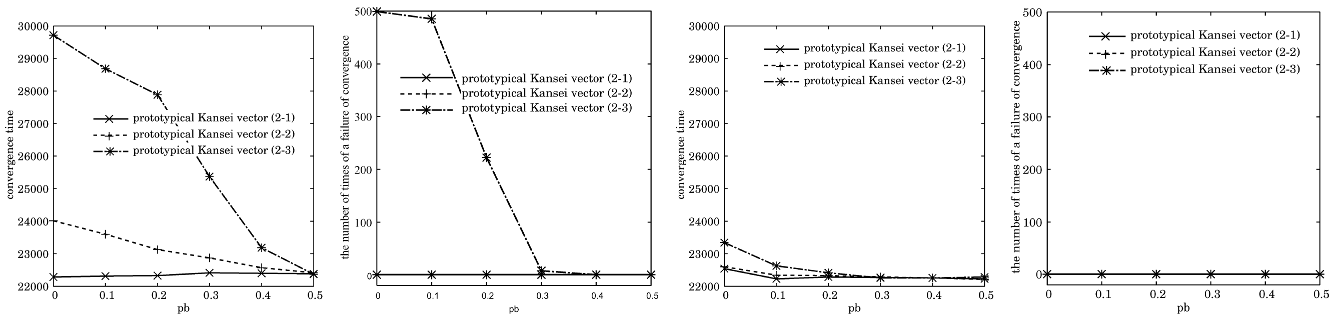
This is the case of two-dimensional vectorized Kansei information.

We can see from Figs. 7 and 8 that the simulation results for the case of random determination of p and those for the case of random determination of $(p1, p2, p3, p4)$ are so similar to each other. In fact, although the dimensions of vectorized Kansei information are different between the cases of random determination of p and $(p1, p2, p3, p4)$, the situations surrounding the peers in both of the cases are similar. That is, if we replace the numbers of -1 and $+1$ appeared in the discussion for the case of random determination of p by the numbers of $(-1, -1)$ and $(+1, +1)$ respectively, we could have the similar discussion here to that for the case of random determination of p.

5.9.3 Case of Kansei-Correlated Determination of p

This is the case of one-dimensional vectorized Kansei information.

According to Fig. 9 (comparing Figs. 9 (a) and 9 (b) for Kansei query forwarding and Figs. 9 (c) and 9 (d) for the one-walker random walk), when using the Kansei-correlated determination of p, the one-walker random walk is superior to or approximately equivalent to the Kansei



(a) Average convergence time. (b) Number of failures to converge. (c) Average convergence time. (d) Number of failures to converge.

Fig. 10 Simulation results obtained using the Kansei-correlated determination of $(p1, p2, p3, p4)$ in the case of two-dimensional vectorized Kansei information. Sub-figures (a) and (b) are for the Kansei query-forwarding method, and sub-figures (c) and (d) are for a one-walker random walk.

query forwarding method in terms of the ability to provide similar search performance for all of the peers. Specifically, when using small values of p_b, the one-walker random walk is superior to the Kansei query forwarding method, and the ability of these two methods becomes closer with increasing values of p_b.

When the difference in the ability to induce convergence between the Kansei query forwarding method and the one-walker random walk was the largest, that is, when using prototypical Kansei vector (3) and $p_b = 0$, every peer has the same tendency during file collection, i.e., the same value of p. In this case, even if a peer making a query randomly chooses other peers, it is likely that the chosen peers have files that the peer making the query desires. Therefore, the one-walker random walk yields good performance with respect to inducing convergence. Meanwhile, in this situation, vectorized Kansei information as search tags works efficiently as search tags. Since the number of "-1" values in the Kansei vectors of the peers is larger than the number of "$+1$" values, and the p value of the peers is proportional to the number of "-1" values, all of the peers need to have approximately the same number of files corresponding to "-1" at any time during the file search in order to achieve quick convergence. However, specific peers are given more opportunities for their files to be downloaded by the Kansei query forwarding method. The reason for this is that these specific peers happened to increase the number of files corresponding to "-1". Consequently, this causes positive feedback to be yielded such that the initial small advantage in the number of files corresponding to "-1" over other peers grows larger and larger with time.

Next, when using prototypical Kansei vector (2), as in the case of using prototypical Kansei vector (3), the Kansei query forwarding method produces the above-mentioned positive feedback to specific peers if the value of p_b is small, because the number of "-1" values in the Kansei vectors of the peers is still larger than the number of "$+1$" values and the p value of the peers is equal to the number of "-1" values. However, when using prototypical Kansei vector (1) and a small value of p_b, since the numbers of "-1" and "$+1$" values in the Kansei vectors of the peers are almost the same, the Kansei query forwarding method does not produce positive feedback to specific peers.

On the other hand, even when using prototypical Kansei vector (3) or (2), the one-walker random walk is expected to provide high search performance that is equivalent for all peers, because each peer has the same tendency with respect to file collection. Therefore, each peer would basically have files that other peers desire. Finally, even when prototypical Kansei vector (3) or (2) is used, if the value of p_b becomes larger, the numbers of "-1" and "$+1$" values in the Kansei vectors of the peers become more similar and randomness is needed in the file search. Therefore, the abilities of the Kansei query forwarding method and the one-walker random walk to induce convergence would become more similar with increasing p_b.

5.9.4 Case of Kansei-Correlated Determination of $(p1, p2, p3, p4)$

This is the case of two-dimensional vectorized Kansei information.

We can see from Figs. 9 and 10 that the simulation results for the case of random determination of p and those for the case of random determination of $(p1, p2, p3, p4)$ are so similar to each other. Similar to the discussion for the cases of random determination of p and $(p1, p2, p3, p4)$, if we replace the numbers of -1 and $+1$ appeared in the discussion for the case of Kansei-correlated determination of p by the numbers of $(-1, -1)$ and $(+1, +1)$ respectively, we could herein have the similar discussion to that for the case of Kansei-correlated determination of p.

5.9.5 Summary of Results

The simulation results are summarized in the following.

- Cases of random determination of p and $(p1, p2, p3, p4)$
  If the Hamming distance between the Kansei vectors of participants is small and the difference between the numbers of "-1" and "$+1$" values or the numbers of

$(-1,-1)$ and $(+1,+1)$ in the Kansei vectors of participants is small , the Kansei query forwarding method has a better ability to induce convergence. Otherwise, there is almost no difference between the Kansei query forwarding method and the one-walker random walk in terms of the ability to induce convergence.

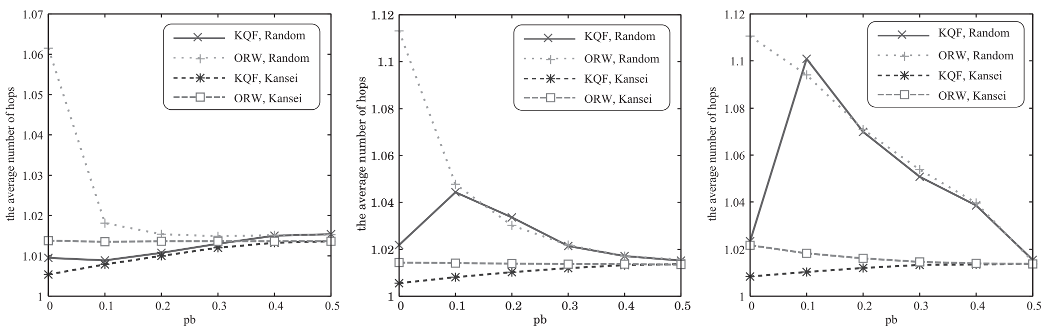
- Cases of Kansei-correlated determination of p and $(p1, p2, p3, p4)$
  If the Hamming distance between the Kansei vectors of participants is small and the difference between the numbers of "-1" and "$+1$" values or the numbers of $(-1,-1)$ and $(+1,+1)$ in the Kansei vectors of participants is large , the one-walker random walk has a better ability to induce convergence. Otherwise, there is almost no difference between the Kansei query forwarding method and the one-walker random walk in terms of the ability to induce convergence.

Although the present paper employed just only one-dimensional and two-dimensional vectorized Kansei information, the simulation results suggest somewhat in general that if the similarity between the Kansei information of the participants is high and the bias of impression to distributed files is small, the Kansei query forwarding method has a better ability to induce convergence and that if the similarity between the Kansei information of participants is high and the bias of impression to distributed files is large, the one-walker random walk has a better ability to induce convergence.

6. Search Speed

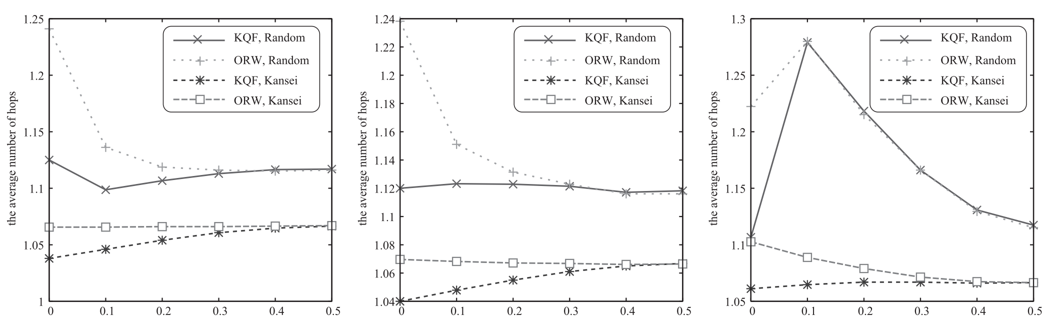
In the previous section, we examined through simulations not search performance for each file search but time required for all of the peers to hold more than or equal to a given number of files. The purpose of this examination was to see how much prepared query forwarding methods can provide equal search performance for the participants with different Kansei information and the tendency of queries. Here, we will compare the query forwarding methods with respect to time required for each file search.

In the previous section, each peer had limitation on the number of times of file downloads and the number of hops



(a) Prototypical Kansei vector (1). (b) Prototypical Kansei vector (2). (c) Prototypical Kansei vector (3).

Fig. 11 The case of one-dimensional vectrized Kansei information. The average number of hops for all of searches on a full-mesh network topology when there is no limitation on the number of times of downloads. The label of "KQF" represents the Kansei query-forwarding, that of "ORW" represents the one-walker random walk, that of "Random" represents the random determination of p, and that of "Kansei" represents the Kansei-correlated determination of p.



(a) Prototypical Kansei vector (2-1). (b) Prototypical Kansei vector (2-2). (c) Prototypical Kansei vector (2-3).

Fig. 12 The case of two-dimensional vectrized Kansei information. The average number of hops for all of searches on a full-mesh network topology when there is no limitation on the number of times of downloads. The label of "KQF" represents the Kansei query-forwarding, that of "ORW" represents the one-walker random walk, that of "Random" represents the random determination of $(p1, p2, p3, p4)$, and that of "Kansei" represents the Kansei-correlated determination of $(p1, p2, p3, p4)$.

allowed for a file search. We here remove this limitation and then investigate the average number of hops for all of searches. The observation period for calculating the average number of hops is from the 1st to the 16,000th searches, at which convergence occurs if no fail of file search. The number of peers is 200 and the network topology is full-mesh. The other simulation configurations are the same as in the previous section.

Figures 11 and 12 show the simulation results for the case of one-dimensional vectorized Kansei information and those for the case of two-dimensional vectorized Kansei information, respectively. The results are the average over 500 independent simulation runs.

According to Figs. 11 and 12, the Kansei query forwarding method is better than or equal to the one-walker random walk in terms of the number of hops. Especially for small values of p_b, the Kansei query forwarding method is better. When the value of p_b is small, the Kansei vectors are similar among all of the peers. That is, what to feel about each of the files is similar among all of the peers. Meanwhile, the Kansei query forwarding method is likely to forward a query to a peer which holds more files with the vectorized Kansei information similar to the query. Since what to feel about each of the files is similar among all of the peers when the value of p_b is small, we can expect that the peer to which the Kansei query forwarding method forwarded a query holds files that a peer making the query wants and also that the Kansei query forwarding method has better search ability than the one-walker random walk in this situation.

These simulation results suggest that when there is no limitation on the number of times of downloads and the number of hops allowed for a file search, the Kansei query forwarding method is always a better choice.

7. Conclusions

In the present paper, we proposed the concept of folksonomical P2P file sharing networks that allow participants (peers) to freely assign structured search tags to their files. We focused on vectorized Kansei (human sensitivity) information as a concrete example of such structured search tags, and then proposed folksonomical P2P file sharing networks using the vectorized Kansei information and the Kansei query forwarding method that enables us to share files having vectorized Kansei information as search tags among participants. The simulation results that when we intend to provide equal search performance for participants with different Kansei information and the tendency of queries, the Kansei query forwarding method and a random-walk-based query forwarding method, for comparison, work effectively in different situations and are complementary. Furthermore, the Kansei query forwarding method is shown, through simulations, to be superior to or equal to the random-walk based one in terms of search speed.

The motivation behind the present work was to enable communication between participants in which information is intermediate. In the proposed folksonomical P2P file sharing networks, participants look for other participants who have a similar sense of file grouping, as well as the files themselves, through a file search process. The simulation results presented in this paper suggest the possibility that participants with similar Kansei information can exchange files with each other depending on situation, and the above-mentioned communication would be one step of the file exchange process described herein. So, in the future work, we need to more mathematically reveal and quantify 'situation' in which participants with similar Kansei information can meet each other, and at the same time, to show the practical usefulness.

Acknowledgment

This study was supported by the Japan Society for the Promotion of Science through a Grant-in-Aid for Scientific Research (S) (18100001).

References

[1] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," IEEE Communications Surveys & Tutorials, vol.7, no.2, pp.72–93, Second Quarter 2005.
[2] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: Scalable peer-to-peer lookup service for internet applications," Proc. ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp.149–160, San Diego, CA, USA, Aug. 2001.
[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," Proc. ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp.161–172, San Diego, CA, USA, Aug. 2001.
[4] A. Rowstron and P. Druschel, "Pastry: Scalable decentralized object location and routing for large-scale peer-to-peer systems," Proc. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pp.329–350, Heidelberg, Germany, Nov. 2001.
[5] B. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for wide-area location and routing," Technical Report UCB CSD-01-1141, U.C. Berkeley, Berkeley, CA, USA, 2001.
[6] Q. Lv, P. Cao, E. Cohen, S. Li, and K. Shenker, "Search and replication in unstructured peer-to-peer networks," Proc. 16th international conference on Supercomputing, pp.84–95, New York, USA, June 2002.
[7] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," Proc. ACM SIGCOMM 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Pittsburgh, PA, USA, Aug. 2002.
[8] A. Mathes, "Folksonomies - cooperative classification and communication through shared metadata."
[9] "Delicious." http://delicious.com/
[10] "Flickr." http://www.flickr.com/
[11] K. Yoshida, T. Kato, and T. Yanaru, "A study of database system with kansei information," Proc. 1999 IEEE International Conference on System, Man, and Cybernetics, pp.253–256, 1999.
[12] H. Takagi and T. Noda, "Media converter with impression preservation using a neuro-genetic approach," Hybrid Intelligent Systems, vol.1, no.1, pp.49–56, 2004.
[13] S. Joseph and T. Hoshiai, "Decentralized meta-data strategies: Effective peer-to-peer search," IEICE Trans. Commun., vol.E86-B, no.6, pp.1740–1753, June 2003.

[14] S. Joseph, "Neurogrid: Semantically routing queries in peer-to-peer networks," Proc. International Workshop on Peer-to-Peer Computing (co-located with Networking 2002), Pisa, Italy, May 2002.
[15] C. Tang, Z. Xu, and S. Dwarkadas, "Peer-to-peer information retrieval using self-organizing semantic overlay networks," Proc. 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM'03), pp.175–186, Karlsruhe, Germany, 2003.
[16] Y. Wang and J. Vassileva, "Trust and reputation model in peer-to-peer networks," Proc. Third International Conference on Peer-to-Peer Computing (P2P 2003), pp.150–157, 2003.
[17] J.A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D.H.J. Epema, M. Reinders, M.R. van Steen, and H.J. Sips, "Tribler: A social-based peer-to-peer system," Concurrency and Computation: Practice & Experience, vol.20, no.2, pp.127–138, 2008.
[18] L. Mekouar, Y. Iraqi, and R. Boutaba, "Personalized recommendations in peer-to-peer systems," 2008 IEEE 17th Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. WETICE 2008, pp.99–104, 2008.
[19] Jun Wang, J. Pouwelse, J. Fokker, A. de Vries, and M. Reinders, "Personalization on a peer-to-peer television system," Multimedia Tools and Applications, vol.36, no.1-2, pp.89–113, Jan. 2008.
[20] H. Nambo, T. Okamine, H. Kimura, M. Nakazawa, and S. Hattori, "An image retrieval system using impressional words based on individual preference," IEEJ Trans. EIS, vol.127, no.11, pp.1937–1946, 2007.

Kei Ohnishi received the B.E., M.E. and D.E. degrees from Kyushu Institute of Design, Japan in 1998, 2000, and 2003, respectively. He worked as a postdoctoral researcher for University of Illinois at Urbana-Champaign, Kyushu Institute of Technology, and Human Media Creation Center / Kyushu. Since 2007, he has been an associate professor at Kyushu Institute of Technology. His research interests include soft computing techniques and P2P networks. He is a member of the IPSJ, and SOFT.

Kaori Yoshida received the B.E., M.E. and D.E. degrees from Kyushu Institute of Technology, Japan in 1994, 1996, and 1999, respectively. She worked at Electrotechnical Laboratory, AIST, MITI, and Kyushu Institute of Technology. Since 2004, she is Associate Professor at Dept. of Artificial Intelligence, Kyushu Institute of Technology, and also collaborating as researcher at Network Design and Research Center. Her research interests include Kansei Information Processing, Human Computer Interaction and Soft Computing. She is active member of IEEE, IPSJ, JSKE, SOFT and BMFSA.

Yuji Oie received B.E., M.E. and D.E. degrees from Kyoto University, Kyoto, Japan in 1978, 1980 and 1987, respectively. From 1980 to 1983, he worked at Nippon Denso Company Ltd., Kariya. From 1983 to 1990, he was with the Department of Electrical Engineering, Sasebo College of Technology, Sasebo. From 1990 to 1995, he was an Associate Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka. From 1995 to 1997, he was a Professor in the Information Technology Center, Nara Institute of Science and Technology. Since April 1997, he has been a Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests include performance evaluation of computer communication networks, high speed networks, and queueing systems. He is a fellow of the IPSJ and a member of the IEEE.