The Journal of Systems and Software 82 (2009) 2023–2037

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Integrating knowledge flow mining and collaborative filtering to support document recommendation

Chin-Hui Lai, Duen-Ren Liu *

Institute of Information Management, National Chiao Tung University, 1001 Ta Hseuh Rd., Hsinchu 300, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Received 11 June 2008
Received in revised form 29 June 2009
Accepted 29 June 2009
Available online 2 July 2009

Keywords:
Knowledge flow
Knowledge flow mining
Knowledge sharing
Document recommendation
Collaborative filtering
Sequential rule mining
Recommender system

ABSTRACT

Knowledge is a critical resource that organizations use to gain and maintain competitive advantages. In the constantly changing business environment, organizations must exploit effective and efficient methods of preserving, sharing and reusing knowledge in order to help knowledge workers find task-relevant information. Hence, an important issue is how to discover and model the knowledge flow (KF) of workers from their historical work records. The objectives of a knowledge flow model are to understand knowledge workers' task-needs and the ways they reference documents, and then provide adaptive knowledge support. This work proposes hybrid recommendation methods based on the knowledge flow model, which integrates KF mining, sequential rule mining and collaborative filtering techniques to recommend codified knowledge. These KF-based recommendation methods involve two phases: a KF mining phase and a KF-based recommendation phase. The KF mining phase identifies each worker's knowledge flow by analyzing his/her knowledge referencing behavior (information needs), while the KF-based recommendation phase utilizes the proposed hybrid methods to proactively provide relevant codified knowledge for the worker. Therefore, the proposed methods use workers' preferences for codified knowledge as well as their knowledge referencing behavior to predict their topics of interest and recommend task-related knowledge. Using data collected from a research institute laboratory, experiments are conducted to evaluate the performance of the proposed hybrid methods and compare them with the traditional CF method. The results of experiments demonstrate that utilizing the document preferences and knowledge referencing behavior of workers can effectively improve the quality of recommendations and facilitate efficient knowledge sharing.

1. Introduction

Organizational knowledge can be used to create core competitive advantages and achieve commercial success in a constantly changing business environment. Hence, organizations need to adopt appropriate strategies to preserve, share and reuse such a valuable asset, as well as to support knowledge workers effectively (Nonaka and Takeuchi, 1995; Polanyi, 1966). Knowledge and expertise are generally codified in textual documents, e.g., papers, manuals and reports, and preserved in a knowledge database. This codified knowledge is then circulated in an organization to support workers engaged in management and operational activities (Brown and Duguid, 2002). Because most of these activities are knowledge-intensive tasks, the effectiveness of knowledge management depends on providing task-relevant documents to meet the information needs of knowledge workers.

In task-based business environments, knowledge management systems (KMSs) can facilitate the preservation, reuse and sharing of knowledge. Moreover, workers may need to obtain task-relevant knowledge to complete a knowledge-intensive task by referencing codified knowledge (documents); For example, based on a task's specifications and the process-context of the task, the *KnowMore* system (Abecker et al., 2000a,b) provides context-aware knowledge retrieval and delivery to support workers' procedural activities. The task-based *K-support* system (Liu et al., 2005; Wu et al., 2005) adaptively provides knowledge support to meet a worker's dynamic information needs by analyzing his/her access behavior or relevance feedback on documents. To help knowledge workers complete multiple tasks, *TaskTracer* (Dragunov et al., 2005) was developed to monitor workers' activities and help them rapidly locate and reuse processes employed previously. However, previous research on task-based knowledge support did not analyze and utilize the flow of knowledge among various types of codified knowledge (documents) to provide effective recommendations about task-relevant documents.

Knowledge flow (KF) research focuses on how KF can transmit, share, and accumulate knowledge when it passes from one team member/process to another. In a workflow situation, work knowledge may flow among workers in an organization, while process

* Corresponding author.
E-mail address: dliu@iim.nctu.edu.tw (D.-R. Liu).

0164-1212/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2009.06.044

knowledge may flow among various tasks (Zhuge, 2002, 2006b; Zhuge and Guo, 2007). Thus, KF reflects the level of knowledge cooperation between workers or processes and influences the effectiveness of teamwork/workflow. Zhuge (2002) proposed a management mechanism for realizing ordered knowledge sharing, and integrated the knowledge flow with the workflow to assist people working in a complex and knowledge-intensive environment. Also, KF plays an important role in academic research, as researchers often devise novel concepts based on previous research reported in the literature (Zhuge, 2006a). However, to the best of our knowledge, there is no systematic method that can flexibly identify KF in order to understand the information needs of workers. Furthermore, conventional KF approaches do not analyze knowledge flow from the perspective of information needs and recommend relevant documents based on the discovered KF.

Knowledge workers normally have various task-needs over time. Moreover, they may need to obtain task-relevant knowledge to complete a task by referencing several types of codified knowledge (documents); and the knowledge in one document may prompt a worker to reference another related document. Based on a worker's referencing behavior, KF can be used to describe the evolution of information needs, preferences, and knowledge accumulated for a specific task. From the perspective of information needs, some knowledge in a KF may have a higher priority for accomplishing a task. For example, before taking a Data Mining course, a student must take courses in Statistics and Database Systems, which represent the fundamental knowledge of Data Mining. Thus, these two courses are significant and have a high priority for the student. Additionally, academic knowledge may flow between different courses and thereby help students accumulate more knowledge. Similarly, the codified knowledge for a task also has different referencing priorities and ordering based on its perceived importance. In other words, important basic knowledge about a task should be referenced first. Therefore, KF can be utilized to provide effective recommendations about task-relevant knowledge to suit workers' information needs for tasks. This issue has not been addressed by previous research.

In an attempt to resolve the limitations of previous research, we propose KF-based recommendation methods for recommending task-related codified knowledge. To adaptively provide relevant knowledge, collaborative filtering (CF), the most frequently used method, predicts a target worker's preference(s) based on the opinions of similar workers. However, the target worker's referencing behavior may change over the period of the task's execution, because his/her information needs may vary. Traditional CF methods only consider workers' preferences for codified knowledge. They neglect the effect of the time factor, i.e., workers' referencing behavior for knowledge over time. To fill this research gap, we propose a KF-based sequential rule method (KSR) that recommends codified knowledge by utilizing the KF-based sequential rules. However, the method is based on the target worker's referencing behavior without considering the opinions of his/her neighbors who may have similar preference for documents. Therefore, to take advantage of the merits of typical CF and KSR methods, we propose hybrid recommendation methods that combine CF and KSR methods to enhance the quality of document recommendation. The hybrid methods consider workers' preferences for codified knowledge, as well as their knowledge referencing behavior, in order to predict topics of interest and recommend task-related knowledge.

The proposed hybrid methods consist of two phases: a KF mining phase and a KF-based recommendation phase. To determine a knowledge worker's referencing behavior, the KF mining phase analyzes his/her historical work records to identify the knowledge flow, i.e., the target worker's information needs. Then, the KF-based recommendation phase selects and recommends documents based on the document preferences and KF-based sequential rules derived from the target worker's neighbors. In other words, the proposed methods trace a worker's information needs by analyzing his/her knowledge referencing behavior for a task over time, and also proactively provide relevant codified knowledge for the worker based on the KFs of the worker's neighbors.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of related works. In Section 3, we describe the knowledge flow-based recommendation framework and knowledge flow model. In Sections 4 and 5, we discuss the knowledge flow mining phase and KF-based recommendation phase respectively. Section 6 details our experimental work, including an evaluation and comparison of our proposed methods and a discussion of the experiment results. Then, in Section 7, we summarize our conclusions and consider future research directions.

2. Background

In this section, we discuss the background of our research, including knowledge flow, information retrieval and task-based knowledge support, document clustering, dynamic programming algorithm, rule-based recommendations, and collaborative filtering.

2.1. Knowledge flow

Knowledge can flow among people and processes to facilitate knowledge sharing and reuse. The concept of knowledge flow has been applied in various domains, e.g., scientific research, communities of practice, teamwork, industry, and organizations (Anjewierden et al., 2005; Kim et al., 2003; Zhuge, 2006a). Scholarly articles represent the major medium for disseminating knowledge among scientists to inspire new ideas (Anjewierden et al., 2005; Zhuge, 2006a). A citation implies that there is knowledge flow between the citing article and the cited article. Such citations form a knowledge flow network that enables knowledge to flow between different scientific projects to promote interdisciplinary research and scientific development.

KM enhances the effectiveness of teamwork by accumulating and sharing knowledge among team members to facilitate peer-to-peer knowledge sharing (Zhuge, 2002). To improve the efficiency of teamwork, Zhuge (Zhuge, 2006b) proposed a pattern-based approach that combines codification and personalization strategies to design an effective knowledge flow network. Kim et al. (2003) proposed a knowledge flow model combined with a process-oriented approach to capture, store, and transfer knowledge. KF in weblogs (blogs) is a communication pattern where the post of one blogger links to that of another blogger to exchange knowledge (Anjewierden et al., 2005). Similarly, knowledge flow in communities of practice helps members share their knowledge and experience about a specific domain to complete their tasks (Rodriguez et al., 2004).

2.2. Information retrieval and task-based knowledge support

Information retrieval (IR) facilitates access to specific items of information (Baeza-Yates and Ribeiro-Neto, 1999; Feldman and Sanger, 2007). The vector space model (Salton and Buckley, 1988) is typically used to represent documents as vectors of index terms, where the weights of the terms are measured by the *tf-idf* approach. *tf* denotes the occurrence frequency of a particular term in the document, while *idf* denotes the inverse document frequency of the term. Terms with higher *tf-idf* weights are used as discriminating terms to filter out common terms. The weight of a term i in a document j, denoted by $w_{i,j}$, is expressed as follows:

$$w_{i,j} = tf_{i,j} \times idf_i = tf_{i,j} \times \left(\log_2 \frac{N}{n} + 1\right), \tag{1}$$

where $tf_{i,j}$ is the frequency of term i in document j, idf_i is measured by $(\log_2 N/n) + 1$, N is the total number of documents in the collection, and n is the number of documents in which term i occurs at least once.

Information retrieval techniques coupled with workflow management systems (WfMS) have been used to support proactive delivery of task-specific knowledge based on the context of tasks within a process (Abecker et al., 2000a,b). For example, the *KnowMore* system (Abecker et al., 2000a,b) provides context-aware delivery of task-specific knowledge. The *Kabiria* system assists knowledge workers with knowledge-based document retrieval by considering the operational context of task-associated procedures (Augusto et al., 1995).

Information filtering with a similarity-based approach is often used to locate knowledge items relevant to the task-at-hand. The discriminating terms of a task are usually extracted from a knowledge item/task to form a task profile, which is used to model a worker's information needs. Holz et al. (2005) proposed a similarity-based approach to organize desktop documents and proactively deliver task-specific information. Liu et al. (2005) proposed a *K-Support* system to provide effective task support for a task-based working environment.

2.3. Document clustering

Document clustering or unsupervised document classification methods are used in many applications. Most methods apply pre-processing steps to the document set and represent each document as a vector of index terms. To cluster similar documents, the similarity between documents is usually measured by the cosine measure (Baeza-Yates and Ribeiro-Neto, 1999; Van RijsBergen, 1979), which computes the cosine of the angle between their corresponding feature vectors. Two documents are considered similar if the cosine similarity value is high. The cosine similarity of two documents, X and Y, is $simcos\ (X,Y) = \frac{\vec{X}\cdot\vec{Y}}{\|\vec{X}\|\|\vec{Y}\|}$, where $\vec{X}$ and $\vec{Y}$ are the feature vectors of X and Y respectively. Documents within a cluster are very similar, while documents in different clusters are very dissimilar.

Agglomerative hierarchical clustering (Johnson, 1967; Kaufman and Rousseeuw, 1990) is a popular document clustering method. In this work, we use the single-link clustering method (Dubes and Jain, 1988; Jain et al., 1999) to cluster codified knowledge (documents). Initially, each document is regarded as a cluster. Next, the single-link method computes the similarity between two clusters, which is equal to the greatest similarity between any document in one cluster and any document in the other cluster. Then, based on the similarity measurement, the two most similar clusters are merged to form a new cluster. The merging process continues until all documents have been merged into one cluster at the top of a hierarchy, or a pre-specified threshold is satisfied (Jain et al., 1999).

2.3.1. Clustering quality

A good clustering method generates clusters that are cohesive and isolated from other clusters. For this reason, the measurement of clustering quality takes both inter-cluster similarity and intra-cluster similarity into account (Chuang and Chien, 2004). Let C be a set of clusters. The inter-cluster similarity between two clusters C_i and C_j, $similarity_A(C_i, C_j)$, is defined as the average of all pairwise similarities between the documents in C_i and C_j; and the intra-cluster similarity within a cluster C_i, $similarity_A(C_i, C_i)$, is defined as the average of all pairwise similarities between documents in C_i. On the basis of the cohesion and isolation of C, the quality measure of C, $CQ(C)$, is defined as:

$$CQ(C) = \frac{1}{|C|}\sum_{C_i \in C} \frac{similarity_A(C_i, \overline{C}_i)}{similarity_A(C_i, C_i)}, \quad \text{where} \quad \overline{C}_i = \cup_{i \neq j} C_j. \tag{2}$$

Note that the smaller the value of $CQ(C)$, the better the quality of the derived set of clusters, C, will be.

2.4. Dynamic programming algorithm for sequence alignment

In this work, each worker's knowledge flow is represented as a sequence. We use sequence alignment techniques to analyze the similarity of workers' knowledge flows, which corresponds to a sequence alignment problem. Such techniques are used to compare or align strings in many application domains, such as biology, speech recognition, and web session clustering. A number of methods can be used for sequence alignment, e.g., the sequence alignment method (SAM) (Hay et al., 2001) and dynamic programming. SAM, also called the string edit distance method (Kruskal, 1983), considers the sequential order of elements in a sequence and then measures the similarity/dissimilarity of sequences. The measurements reflect the operations necessary to equalize the sequences by computing the costs of deleting and inserting unique elements as well as the costs of reordering common elements (Hay et al., 2001; Mannila and Ronkainen, 1997). In addition, Charter et al. (2000) proposed a dynamic programming algorithm that solves the sequence alignment problem efficiently.

The algorithm consists of three steps: initialization, *FindScore* and *FindPath* (Charter et al., 2000; Oguducu and Ozsu, 2006). The first step creates a dynamic programming matrix with $N + 1$ columns and $M + 1$ rows, where N and M correspond to the sizes of the sequences to be aligned. One sequence is placed at the top of the matrix and the other is placed on the left-hand side of the matrix. There is a gap at the end of each sequence to allow calculation of the alignment score. The *FindScore* step calculates the two-dimensional alignment score of sequences. If two aligned sequences have an identical matching in the same column, the column is given a positive score s (e.g., +1 or +2); but if the values in a column are mismatches, the score s is zero or negative (e.g., 0, -1 or -2). In addition, if a column contains a gap, it is given a penalty score w (e.g., 0, -1 or -2). Therefore, starting from the bottom right-hand corner, each position in the dynamic programming matrix is given the maximal score M_{ij}. For each position in the matrix, M_{ij} is defined as follows:

$$M_{ij} = Maximum\{(M_{i-1,j-1} + s_{ij}), (M_{i,j-1} + w), (M_{i-1,j} + w)\}, \tag{3}$$

where i is the row number, j is the column number, s_{ij} is the match/mismatch score, and w is the penalty score. The third step, *FindPath*, determines the actual KF alignment that derives the maximal score. It traverses the matrix from the destination point (top left-hand corner) to the starting point (bottom right-hand corner) to find an optimal alignment path in order to determine the maximal alignment score δ. We calculate the flow similarity based on the maximal alignment score. The details are given in Section 5.1.

2.5. Collaborative filtering recommendation

Collaborative filtering (CF) is a well-known approach for recommender systems: GroupLens (Konstan et al., 1997), Ringo (Shardanand and Maes, 1995), Siteseer (Rucker and Polanco, 1997), and Knowledge Pump (Glance et al., 1998). CF recommends items, e.g., products, movies, and documents, based on the preferences of people who have the same or similar interests to those of the target user (Breese et al., 1998; Liu et al., 2008; Liu and Shih, 2005). The CF approach involves two steps: neighborhood formation and prediction. The neighborhood of a target user is selected according to his/her similarity to other users, and is computed by Pearson correlation coefficient or the cosine measure. Either the k-NN (nearest neighbor) approach or a threshold-based approach is used to choose n users that are most similar to the target user. Here, we use the k-NN approach. In the prediction step, the predicted rating is calculated from the aggregated weights of the selected n nearest neighbors' ratings, as shown in Eq. (4):

$$P_{u,j} = \overline{r_u} + \frac{\sum_{i=1}^{n} w(u,i)(r_{i,j} - \overline{r_i})}{\sum_{i=1}^{n} |w(u,i)|}, \tag{4}$$

where $P_{u,j}$ denotes the prediction rating of item j for the target user u; $\overline{r_u}$ and $\overline{r_i}$ are the average ratings of user u and user i, respectively; $w(u,i)$ is the similarity between target user u and user i; $r_{i,j}$ is the rating of user i for item j; and n is the number of users in the neighborhood.

Similar to the PCF method, the item-based collaborative filtering (ICF) algorithm (Linden et al., 2003; Sarwar et al., 2001) analyzes the relationships between items (e.g., documents) first, rather than the relationships between users. Then, the item relationships are used to compute recommendations for workers indirectly by finding items that are similar to other items the worker has accessed previously. Thus, the prediction for an item j for a user u is calculated by the weighted sum of the ratings given by the user for items similar to j and weighted by the item similarity, as shown in Eq. (5).

$$p_{u,j} = \frac{\sum_{m=1}^{n} w(j,m) \times r_{u,m}}{\sum_{m=1}^{n} |w(j,m)|}, \tag{5}$$

where $p_{u,j}$ represents the predicted rating of item j for user u; $w(j,m)$ is the similarity between two items j and m; and $r_{u,m}$ denotes the rating of user u for item m. A number of methods can be used to determine the similarity between items e.g., the cosine-based similarity, correlation-based similarity, and adjusted cosine similarity methods. Since the adjusted cosine similarity method performs better than the others (Sarwar et al., 2001), we use it as the similarity measure for the ICF method. The adjusted cosine similarity between two items i and j is given by Eq. (6).

$$sim(i,j) = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_u)^2}}, \tag{6}$$

where $r_{u,i}/r_{u,j}$ is the rating of item i/j given by user u; and $\bar{r}_u$ is the average item rating of user u.

2.6. Rule-based recommendations

Association rule mining (Agrawal et al., 1993; Agrawal and Srikant, 1994; Yun et al., 2003) is a widely used data mining technique that generates recommendations in recommender systems. An association rule describes the relationships between items, such as products, documents, or movies, based on patterns of co-occurrence across transactions. The Apriori algorithm (Agrawal et al., 1993; Agrawal and Srikant, 1994) is usually employed to identify such rules. Two measures, support and confidence, are used to indicate the quality of an association rule (Agrawal et al., 1993). The discovered rules should satisfy two user-defined requirements, namely minimum support and minimum confidence.

To improve the quality of traditional CF, Cho et al. (2005) proposed a sequential rule-based recommendation method that considers the evolution of customers' purchase sequences. Transactions are clustered into a set of q transaction clusters, $C = \{C_1, C_2, \ldots, C_q\}$, where each C_j is a subset of transactions. Each customer's transactions over l periods are then transformed into transaction clusters as a behavior locus, $L_i = \langle C_{i,T-l+1}, \ldots C_{i,T-1}, C_{i,T}\rangle$, where $C_{i,T-k} \in C$, $k = 1,2, \ldots, l-1$, $l \geqq 2$. Finally, sequential purchase patterns are extracted from the behavior locus of customers by time-based association rule mining to keep track of customers' preferences during l periods, with T as the current (latest) period. A sequential rule is expressed in the form $C_{T-l+1}, \ldots, C_{T-1} \Rightarrow C_T$, where C_T represents the customers' purchase behavior in period T. If a target customer's purchase behavior prior to period T was similar to the conditional part of the rule, then it is predicted that his/her purchase behavior in period T will be C_T. Accordingly, C_T is used to recommend products to the target customer in T.

3. Knowledge flow-based recommendation framework

In this work, we propose three hybrid recommendation methods based on knowledge flow (KF), which is a sequence of codified knowledge (documents) or topics referenced by a worker during a task's execution. KF represents a worker's information needs and the evolution of knowledge requirements, and is identified by analyzing a worker's work log. To support workers effectively, our methods consider workers' preferences as well as their referencing behavior in order to recommend task-related knowledge. During the recommendation phase, the user-based collaborative filtering (CF) is used to predict a target worker's preferences based on the opinions of similar workers, while the item-based collaborative filtering (Sarwar et al., 2001) is used to predict a document based on the targets worker's interests on its similar items (documents). However, the limitation of these traditional CF methods is that they only consider workers' preferences for codified knowledge and neglect workers' referencing behavior. A worker's referencing behavior may change during the task's execution to suit his/her current information needs. To address this issue, we propose a KF-based sequential rule method that improves the recommendation quality by tracking workers' referencing behavior based on sequential rules. However, this method does not consider the opinions of the target worker's neighbors who have similar preferences for documents. To overcome the limitations of CF and KF-based sequential rule methods, we combine the advantages of the two approaches and propose three hybrid recommendation methods that integrate KF mining, KF-based sequential rule mining and CF techniques to enhance the quality of recommendations.

3.1. Recommendation processes based on the knowledge flow model

The proposed recommendation methods are illustrated in Fig. 1. Our methods consist of two phases, a knowledge flow mining phase and a KF-based recommendation phase. The first phase identifies the worker's knowledge flow from the large amount of knowledge in the worker's log. Then, the second phase recommends codified knowledge to the target worker by using the proposed recommendation methods.

In the knowledge flow mining phase, KFs are identified from the task requirements and the referencing behavior of workers recorded in their logs. As tasks are performed at various times, each knowledge worker requires different kinds of knowledge to achieve a goal or complete a task. This phase involves three steps: document profiling, document clustering, and knowledge flow extraction. In the first step, each document is represented as a document profile, which is an n-dimensional vector comprised of significant terms and their weights. Then, based on the document profiles, documents with higher similarity measures are grouped in clusters by the hierarchical clustering method. In the third step, topic-level and codified-level KFs are generated from the document clustering results. A topic-level KF is expressed as a sequence of topics referenced by a worker, while a codified-level

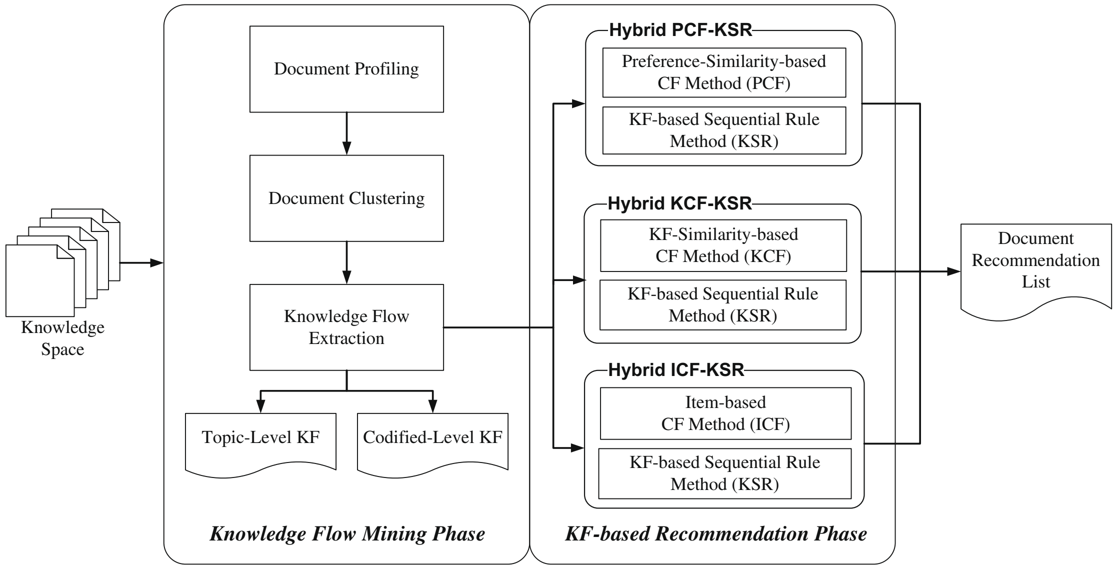
Knowledge Space
Document Profiling
Document Clustering
Knowledge Flow Extraction
Topic-Level KF
Codified-Level KF
Knowledge Flow Mining Phase
Hybrid PCF-KSR
Preference-Similarity-based CF Method (PCF)
KF-based Sequential Rule Method (KSR)
Hybrid KCF-KSR
KF-Similarity-based CF Method (KCF)
KF-based Sequential Rule Method (KSR)
Hybrid ICF-KSR
Item-based CF Method (ICF)
KF-based Sequential Rule Method (KSR)
KF-based Recommendation Phase
Document Recommendation List

Fig. 1. Document recommendation based on knowledge flows.

KF is represented as a sequence of codified knowledge accessed by a worker. Further details are given in Section 4.

The proposed hybrid recommendation methods combine a KF-based sequential rule (KSR) method with a user-based/item-based collaborative filtering (CF). The KSR method is regarded as the core process of the proposed hybrid methods. In the KSR method, workers with similar KFs to that of the target worker are deemed neighbors of the target worker and their knowledge referencing behavior patterns are identified by a sequential rule mining method. Based on the discovered sequential rules and the neighbors' KFs, relevant topics and codified knowledge are recommended to the target worker to support the task-at-hand. Moreover, by considering workers' preferences for codified knowledge, the CF method makes recommendations to the target worker based on the opinions of similar workers. Three approaches are used to find similar workers to the target worker. The preference-similarity-based CF method (PCF) chooses workers with similar preferences, while the KF-similarity-based CF method (KCF) chooses workers with similar KFs. Different from these two user-based methods, the item-based CF method predicts a document rating based on its similar documents that have been rated by a target user. To adaptively and proactively recommend codified knowledge, we consider workers' referencing behavior as well as their preferences for codified knowledge. Therefore, three hybrid recommendation methods are used in the KF-based recommendation phase: (1) a hybrid of PCF and KSR (PCF–KSR), (2) a hybrid of KCF and KSR (KCF–KSR), and (3) a hybrid of ICF and KSR (ICF–KSR). Further details are given in Section 5. In the following sections, we describe our methods in detail, including the knowledge flow model, the knowledge flow mining phase and the KF-based recommendation phase.

3.2. Knowledge flow model

In a knowledge-intensive and task-based environment, workers may need to access a large number of documents (codified knowledge) to accomplish a task. From the perspective of information needs, a worker's knowledge flow (KF) represents the evolution of his/her information needs and preferences during a task's execution. Workers' KFs are identified by analyzing their knowledge referencing behavior based on their historical work logs, which contain information about previously executed tasks, task-related documents and when the documents were accessed.

A KF consists of two levels: a codified-level and a topic-level, as shown in Fig. 2. The knowledge in the codified-level indicates the knowledge flow between documents based on the access time. In most situations, the knowledge obtained from one document prompts a knowledge worker to access the next relevant document (codified knowledge). Hence, the task-related documents are sorted by their access time to obtain a document sequence as the codified-level KF.

Documents with similar concepts can be grouped together automatically to form a topic-level abstraction of knowledge. Note that each topic may contain several task-related documents. The codified-level KF can be abstracted to form a topic-level KF, which represents the transitions between various topics. Since the task knowledge in the topic-level may flow among topics, it could prompt the worker(s) to retrieve knowledge from the next related topic. Formally, we define knowledge flow as follows.

Definition 1 (*Knowledge Flow (KF)*). Let a worker's knowledge flow be $KFlow_w^v = \{TKF_w^v, CKF_w^v\}$, where TKF_w^v is the topic-level KF of the worker w for a task v, and CKF_w^v is his/her codified-level KF for the task v.

Definition 2 (*Codified-Level KF*). A codified-level KF is a time-ordered sequence arranged according to the access times of the documents it contains. Thus, it is defined as $CKF_w^v = \langle d_w^{t_1}, d_w^{t_2}, \ldots, d_w^{t_f} \rangle$ and $t_1 < t_2 < \cdots < t_f$, where $d_w^{t_j}$ denotes the document that the worker w accessed at time t_j for a specific task v. Each document can be represented by a document profile, which is an n-dimensional vector containing weighted terms that indicate the key content of the document.

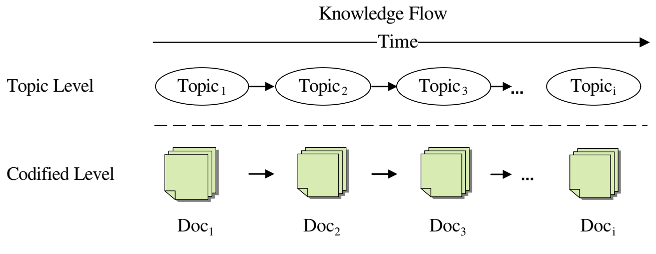

Fig. 2. The two levels of a knowledge flow.

Definition 3 (*Topic-Level KF*). A topic-level KF is a time-ordered topic sequence derived by mapping documents in the codified-level KF to corresponding topics. Thus, it is defined as $TKF_w^v = \langle TP_w^{t_1}, TP_w^{t_2}, \ldots, TP_w^{t_f} \rangle, t_1 < t_2 < \cdots < t_f$, where $TP_w^{t_j}$ denotes the corresponding topic of the document that worker w accessed at time t_j for a specific task v. Each topic is represented by a topic profile, which is an n-dimensional vector containing weighted terms that indicate the key content of the topic.

4. Knowledge flow mining phase

The objective of the knowledge flow (KF) mining phase is to identify the KF of each knowledge worker. In this section, we describe how the KF mining method identifies KFs from workers' log. This phase consists of three steps: document profiling, document clustering and KF extraction, which we discuss in the following subsections.

4.1. Document profiling and document clustering

Two profiles, a document profile and a topic profile, are used to represent a worker's KF. A document profile can be represented as an n-dimensional vector composed of terms and their respective weights derived by the normalized *tf-idf* approach based on Eq. (1). Based on the term weights, terms with higher values are selected as discriminative terms to describe the characteristics of a document. The document profile of d_j is comprised of these discriminative terms. Let the document profile be $DP_j = \langle dt_{1j} : dtw_{1j}, dt_{2j} : dtw_{2j}, \ldots, dt_{nj} : dtw_{nj} \rangle$, where dt_{ij} is the term i in d_j and dtw_{ij} is the degree of importance of a term i to the document d_j, which is derived by the normalized *tf-idf* approach. The document profiles are used to measure the similarity of the documents.

We adopt the single-link hierarchical clustering method (Jain et al., 1999) to group documents with similar profiles into clusters by using the cosine measure to calculate the similarity between the profiles of two documents. The single-link method computes the cluster similarity between two clusters C_r and C_t by $\underset{d_i \in C_r, d_j \in C_t}{max}\{simcos(d_i, d_j)\}$ (Zhao et al., 2005), and then merges the two most similar clusters into a single cluster. The similarity computation and cluster combination steps are repeated until the similarity of the most similar pair of clusters is lower than a prespecified threshold value. Different clustering results can be obtained by setting different threshold values. We adjust the threshold value systematically and use the quality measure described in Section 2.3.1 to evaluate each clustering result. Then, we take the one with the best quality measure as our clustering result. Note that a cluster represents a topic set and has a topic profile (derived from the document cluster) that describes the features of the topic.

4.1.1. Topic profile

Documents in the same cluster contain similar content and form a topic set. The key features of the cluster are described by a topic profile, which is derived from the profiles of documents that belong to the cluster. Let $TP_x = \langle tt_{1x}{:}ttw_{1x}, tt_{2x}{:}ttw_{2x}, \ldots, tt_{nx}{:}dtw_{nx} \rangle$ be the profile of a topic (cluster) x, where tt_{ix} is a topic term and ttw_{ix} is the weight of the topic term. In addition, let D_x be the set of documents in cluster x. The weight of a topic term is determined by Eq. (7) as follows:

$$ttw_{ix} = \frac{\sum_{j \in D_x} dtw_{ij}}{|D_x|}, \tag{7}$$

where dtw_{ij} is the weight of term i in document j, and $|D_x|$ is the number of documents in cluster x. The weight of a topic term is obtained from the average weight of the terms in the document set.

4.2. Knowledge flow extraction

In this section, we describe the method used to extract a worker's KF from his/her data log when performing a task. We define a task as a unit of work, which denotes either a previously executed (i.e., historical) task or the current task. When performing a task in a knowledge-intensive and task-based environment, a worker usually requires a large amount of task-related knowledge to accomplish the task. By analyzing a worker's referencing behavior for a specific task, the corresponding knowledge flow of the task is derived by the knowledge flow extraction method. Note that if a worker performs more than one task, more than one knowledge flow will be extracted. For a specific task, the method derives two kinds of KF, *codified-level KF* and *topic-level KF*, to represent the worker's information needs for the task.

4.2.1. Codified-level knowledge flow

The codified-level KF is extracted from the documents recorded in the worker's work log. In most situations, workers are motivated to access a document about a specific task because of knowledge derived from other documents. The documents are arranged according to the times they were accessed, and a document sequence, i.e., a codified-level KF, is obtained. The order of documents in the sequence is subjective, since it is determined by the worker. In other words, each worker has his/her own codified-level KF, which represents his/her knowledge accumulation process for a specific task at the codified-level.

4.2.2. Topic-level knowledge flow

The topic-level KF is derived by mapping documents in the codified-level KF of a specific task into corresponding clusters and is represented by a topic sequence. In the previous step, documents with similar content were grouped into clusters. We use the document clustering results to map the documents in the codified-level KF into topics (clusters) in order to compile the topic-level KF. Since the codified-level KF is the basis of the topic-level KF, the knowledge in the latter is an abstraction of the former, and indicates how knowledge flows among various topics. A topic in the topic-level KF may be duplicated because the worker may read about the same topic frequently to obtain essential knowledge while executing a task.

5. KF-based recommendation phase

The KF-based recommendation phase consists of three hybrid recommendation methods: (1) PCF and KSR (PCF–KSR), (2) KCF and KSR (KCF–KSR), and (3) ICF and KSR (ICF–KSR), as shown in Fig. 1. We note that PCF denotes the preference-similarity-based CF method; KCF denotes the KF-similarity-based CF method; ICF denotes the item-based CF method; and KSR denotes the KF-based sequential rule method. To adaptively recommend documents, both the PCF method and the KCF method select neighbors based on the similarity of preferences, while the ICF method chooses similar documents for a document based on their preferences given by a target user. The three methods differ in the way they compute the similarity between workers' preferences to select the target worker's neighbors. The PCF method (traditional CF) uses preference ratings to compute the similarity, while the KCF method uses workers' KFs to derive the similarity. The ICF method applies similarity measure to evaluate the similarity between two items (i.e., documents), rather than the similarity between two workers. The proposed KSR method traces workers' knowledge referencing behavior by using the KF-based sequential rules. The proposed hybrid recommendation methods take advantage of the merits of the KSR, PCF, KCF and ICF methods.

5.1. Identifying similar workers based on their knowledge flows

To find a target worker's neighbors, his/her topic-level KF is compared with those of other workers to compute the similarity of their KFs. The resulting similarity measure indicates whether the KF referencing behavior of two workers is similar. In this work, we regard each knowledge flow as a sequence. Since comparing knowledge flows is very similar to aligning sequences, the sequence alignment method (SAM) (Hay et al., 2001) and the dynamic programming approach (Charter et al., 2000; Oguducu and Ozsu, 2006) can be used to measure the similarity of two KF sequences.

To determine which of the two methods would be more appropriate for comparing workers' knowledge flows, we applied both methods in our experiments and found that dynamic programming is better than SAM. Therefore, we employ the dynamic programming algorithm (Charter et al., 2000; Oguducu and Ozsu, 2006) to measure the similarity of workers' knowledge flows.

Unlike the sequence alignment problem, a worker's KF contains task-related documents. Thus, we have to consider the sequential order of topics in a knowledge flow, as well as the worker's aggregated profile, which accumulates the task-related documents based on the times they were accessed during the task's execution. We propose a hybrid similarity measure, comprised of the KF alignment similarity and the aggregated profile similarity, to evaluate the similarity of two workers' KFs, as shown in Eq. (8).

$$sim(TKF_i^v, TKF_j^l) = \alpha \times sim_a(TKF_i^v, TKF_j^l) + (1-\alpha) \times sim_P(AP_i^v, AP_j^l), \tag{8}$$

where $sim_a(TKF_i^v, TKF_j^l)$ represents the KF alignment similarity between worker i and worker j who execute task v and task l, respectively; TKF_i^v/TKF_j^l is the topic-level KF of worker i/j for task v/l; $sim_p(AP_i^v, AP_j^l)$ represents the aggregated profile similarity of two workers' KFs; AP_i^v/AP_j^l is the aggregated profile of worker i/j for task v/l; and α is a parameter used to adjust the relative importance of the two types of similarity.

The KF alignment similarity is based on the topic sequence and topic coverage, while the aggregated profile similarity is based on the aggregated profiles derived from the profiles of referenced documents in the KFs. Note that the KF alignment similarity considers the topic sequence in the KF without considering the content of workers' profiles; while the aggregated profile similarity considers the content of profiles without considering the topic sequence in the KF. By linearly combining these two similarities, we can balance the tradeoff between KF alignment and the aggregated profile. We discuss the rationale behind these two similarity measures next.

5.1.1. KF alignment similarity

The KF alignment similarity is comprised of two parts: the KF alignment score, which measures the topics in sequence; and the join coefficient, which estimates the topic's coverage in two compared topic-level KFs. We modify the sequence alignment method (Charter et al., 2000) to derive the KF alignment score. In addition to computing the sequence alignment score, we estimate the overlap of the topics in two compared topic-level KFs by using the Dice's coefficient (Van RijsBergen, 1979). The rationale is that if the topic overlap is high, the KF alignment similarity of the two compared KFs will also be high. In other words, the two compared KFs will be very similar. The KF alignment similarity, $sim_a(TKF_i^v, TKF_j^l)$, is defined as follows:

$$sim_a(TKF_i^v, TKF_j^l) = Norm(\eta) \times \frac{2 \times |TPS_i^v \cap TPS_j^l|}{|TPS_i^v| + |TPS_j^l|}, \tag{9}$$

where TKF_i^v/TKF_j^l denotes the topic-level KF of worker i/worker j for task v/task l; η is the KF alignment score; *Norm* is a normalization function used to transform the value of η into a number between 0 and 1; TPS_i^v and TPS_j^l are the sets of topics in TKF_i^v and TKF_j^l, respectively; $TPS_i^v \cap TPS_j^l$ is the intersection of topics common to TKF_i^v and TKF_j^l; and $|TPS_i^v|$ and $|TPS_j^l|$ represent the number of topics in TKF_i^v and TKF_j^l respectively. The KF alignment score, which is based on the sequence alignment method (Oguducu and Ozsu, 2006), is defined in Eq. (10):

$$\eta = \frac{\delta}{m_s \times \xi}, \tag{10}$$

where δ is the maximal alignment score derived by the dynamic programming approach, m_s is the identical matching score (+2), and ξ is the length of the aligned KF. To obtain the maximal alignment score δ, we set the matching score m_s, the mismatching score m_d and the gap penalty score m_g to +2, −1 and −2, respectively in the dynamic programming approach (Charter et al., 2000) discussed in Section 2.4. The maximum value of η is 1 if the two compared KFs are exactly the same. On the other hand, the value of η is negative if most of topics in the two compared KFs do not match. Thus, the value of η may range from a negative value to 1. To alter the range of the KF alignment score, the value of η is transformed into a value in the range $[0, 1]$ by the normalization function. The normalized KF alignment score $Norm(\eta)$ is then used to calculate the KF alignment similarity.

5.1.2. Aggregated profile similarity

The aggregated profile similarity, defined as $sim_p(AP_i^v, AP_j^l)$, computes the similarity of two workers' KFs based on their aggregated profiles, which are derived from the profiles of documents they have referenced; AP_i^v and AP_j^l are the respective vectors of the aggregated profiles of workers i/j for task v/l. We use the cosine formula to calculate the similarity between two aggregated profiles. The value of the similarity score ranges from 0 to 1. The aggregated profile of a worker i for task v is defined as

$$AP_i^v = \sum_{t=1}^{T} tw_{t,T} \times DP_t^v, \tag{11}$$

where $tw_{t,T}$ is the time weight of the document referenced at time t in the KF; T is the index of the times the worker accessed the most recent documents in his KF; and DP_t^v is the profile of the document referenced by worker i at time t for task v. The aggregation process considers the time decay effect of the documents. Each document profile is assigned a time weight according to the time it was referenced. Thus, higher time weights are given to documents referenced in the recent past. The time weight of each document profile is defined as $tw_{t,T} = \frac{t-St}{T-St}$, where St is the start time of the worker's KF.

5.2. KF-based sequential rule method

The KF-based sequential rule method (KSR) considers the referencing behavior of neighbors whose KFs were very similar before time T, and then recommends documents at time T for the target worker. Fig. 3 provides an overview of the KSR method. To determine the similarity of various topic-level KFs, the target worker's KF is compared with those of other workers by measuring their KF similarity, as discussed in Section 5.1. Workers with similar KFs to that of the target worker are regarded as the latter's neighbors and their topic-level KFs are used to discover frequent knowledge referencing behavior by applying sequential rule mining to the target worker's referencing behavior. The discovered sequential rules with high degrees of rule matching are selected to recommend topics at time T. Documents belonging to the recommended topics have a high priority of being recommended.

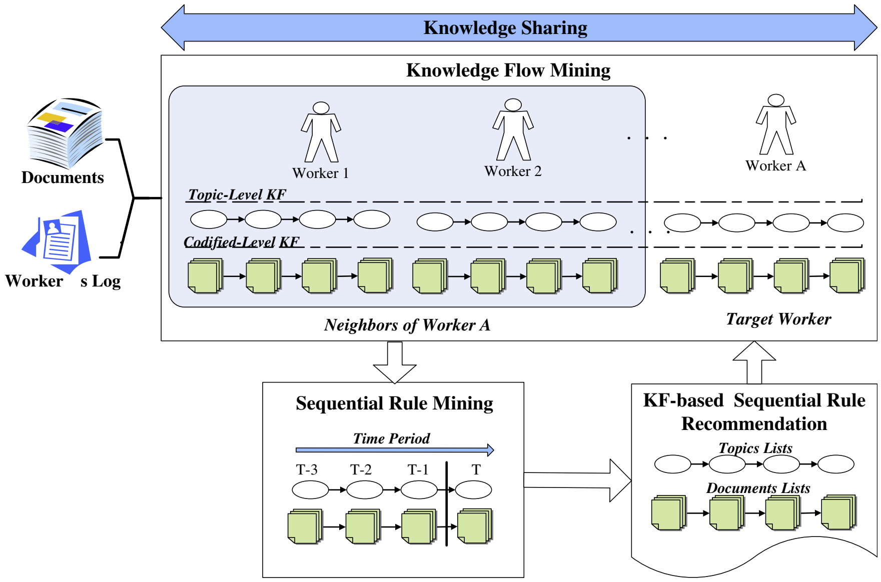

Fig. 3. An overview of the KSR method.

The KSR recommendation method involves four steps: identifying similar workers, mining their knowledge referencing behavior, identifying the target worker's knowledge referencing behavior, and document recommendation.

5.2.1. Mining knowledge referencing behavior

Knowledge workers with similar referencing behavior (high similarities) of the target worker are regarded as neighbors of the target worker. We modify the association rule mining method (Agrawal et al., 1993; Agrawal and Srikant, 1994) and sequential pattern mining method (Agrawal and Srikant, 1995) to discover topic-level sequential rules from the neighbors' topic-level KFs. The extracted rules can be used to keep track of the referenced topics among workers with similar referencing behavior. Let R_y be a sequential rule, as defined in Eq. (12).

$$R_y : g_{y,T-s}, \ldots, g_{y,T-1} \Rightarrow g_{y,T}(Support_y, Confidence_y) \qquad (12)$$

where $g_{y,T-f} \in TPS$; $f = 0$ to s; and *TPS* is a set of all topics.

The conditional part of the sequential rule is $\langle g_{y,T-s}, \ldots, g_{y,T-1} \rangle$, and the consequent part is $g_{y,T}$. The items that appear in the rules are topics extracted from the neighbors' topic-level KFs (*TKF*). The support and confidence values, $Support_y$ and $Confidence_y$, are used to evaluate the importance of rule R_y. We use the support and confidence scores to measure the degree of match between the referencing behavior and the conditional part of a rule for a target worker, as illustrated in the third step. Note that if the knowledge referencing behavior of the target worker is similar to the conditional part of R_y, then the topic predicted for him/her at *T* will be $g_{y,T}$.

5.2.2. Identifying the knowledge referencing behavior of the target worker

This step identifies the target worker's knowledge referencing behavior by matching his/her KF with the sequential rules discovered in the previous step. Specifically, the rules are matched with the topic-level KF of the target worker to predict the topics required at time *T*. We set a knowledge window on the KF before time *T*. The size of the window is determined by the user. Let $KW_u = \langle TP_u^{T-s}, TP_u^{T-s+1}, \ldots, TP_u^{T-1} \rangle$ be the knowledge window for the topic-level KF of a target worker *u* before time *T*. Note that TP_u^{T-f} is the topic referenced by *u* at time $T-f$, $f = 1, \ldots, s$. The knowledge window KW_u covers several topics previously referenced by the target worker and arranged in time order. The steps of sequential rule matching are as follows.

Step 1. Set a knowledge window KW_u

The reference time of topics in the window may range from $T-s$ to $T-1$, where *s* is the window size determined by the worker. The referencing behavior within the knowledge window is then compared with the sequential rules extracted from the KFs of the target worker's neighbors (Step 3).

Step 2. Generate topic subsequences and compare them with the knowledge window

All generated rules are compared with the given knowledge window to obtain the matching scores of rules. A sequential rule may partially or fully match a knowledge window. To identify sequential rules that match the target worker's referencing behavior, we consider all partial matches of the rules. Therefore, all possible topic subsequences are generated from the conditional part of the rule first.

The topic subsequences are enumerated according to the topic order in the conditional part of a rule. Let $TS_y^k = \langle TP_y^{k_1}, \ldots, TP_y^{k_i}, \ldots, TP_y^{k_m} \rangle$ be a topic subsequence in the conditional part of a sequential rule *y*, and let $TP_y^{k_i}$ be a topic with the index position k_i in the sequence TS_y^k. In addition, let KW_u be a knowledge window in a worker's KF, and let TP_u^{hj} be a topic with the index position h_j in the sequence KW_u. Then, each topic subsequence of a rule is examined by checking whether it exists in the knowledge window.

Instead of using identical matches, all the topics in a topic subsequence are compared with those in the knowledge window by using topic similarities to determine their matches. The characteristics of a KF are different from those of a general sequence, because a topic in a KF is composed of abstract knowledge concepts. Rather than using the identical match method, we use the topic similarity, i.e., $simcos(TP_y^{k_i}, TP_u^{h_j})$, to determine if two topics match. That is, they match if their similarity is greater than the user-specified threshold θ.

We define a similarity matching score to compare a topic subsequence with a knowledge window. A topic subsequence TS_y^k

matches the knowledge window KW_u, if their corresponding topic similarities are larger than the user-defined threshold, i.e. $simcos(TP_y^{k_1}, TP_u^{h_1}) > \theta, simcos(TP_y^{k_2}, TP_u^{h_2}) > \theta, \ldots, simcos(TP_y^{k_m}, TP_u^{h_m}) > \theta$, where integers $k_1 < k_2 < \cdots < k_m$, $h_1 < h_2 < \cdots < h_m$, and θ is the user-defined threshold. The similarity matching score is the summation of the topic similarities, as defined in Eq. (13).

$$SM_{TS_y^k, KW_u} = \sum_{i=1}^{m} simcos(TP_y^{k_i}, TP_u^{h_i}), \tag{13}$$

Step 3. Find the matching degree of a sequential rule

Given the similarity matching scores of all topic subsequences extracted from a sequential rule, we choose the subsequence with the highest score to compute the matching degree of the rule. The matching degree is defined as follows:

$$RMD_{R_y, KW_u} = \max_{k=1,\ldots,q} \{SM_{TS_y^k, KW_u}\} \times Support_y \times Confidence_y, \tag{14}$$

where RMD_{R_y, KW_u} is the matching degree of rule R_y and KW_u of the target worker *u;* and $\max\limits_{k=1,\ldots,q} \{SM_{TS_y^k, KW_u}\}$ is the highest similarity matching score of all topic subsequences of sequential rule *y*. The matching degree is used to identify the sequential rules qualified to recommend topics at time *T*.

Step 4. Choose sequential rules for recommendation

A sequential rule with a high matching degree means that the referencing behavior of the target worker matches the conditional part of the rule, so the consequent part of the rule can be selected as a predicted topic for the target worker at time *T*. Hence, the Top-*N* approach can be used to derive a set of predicted topics by selecting *N* rules with the highest matching degree scores.

5.2.3. Document recommendation

The KSR method predicts a document rating based on sequential rules derived from the KFs of a target worker's neighbors. Let KNB_u^v be a set of neighbors of target worker *u* for a task *v*, selected according to the KF similarity (using Eq. (8)). The sequential rules derived from KNB_u^v with high degrees of rule matching are selected to recommend topics for the target worker at time *T*. However, the referencing behavior of some workers in KNB_u^v may not match the selected sequential rules. Therefore, we apply the sequential rule matching method discussed in Section 5.2.2 to compare the KFs of workers in KNB_u^v with the selected sequential rules. If a worker's KF matches a selected sequential rule, that worker's referencing behavior conforms to the sequential rule, and can therefore be used to make recommendations based on the selected sequential rules. The reason for checking the KFs of workers in KNB_u^v is to identify neighbors whose referencing behavior conforms to the selected sequential rule.

For a task *v*, let $KNBR_u^v$ denote the neighbors in KNB_u^v whose KFs are very similar to the target worker's KF and whose referencing behavior matches the selected sequential rules. In addition, let *RTS* be a set of recommended topics derived from the consequent parts of the recommended sequential rules; τ be a recommended topic, where $\tau \in RTS$; and the topic of a document *d* be τ. Based on the KFs of the neighbors in $KNBR_u^v$, the predicted rating of a document *d* belonging to the recommended topic τ for the target worker *u* is calculated by Eq. (15):

$$\hat{p}_{u,d,\tau}^{v} = \bar{r}_{u,\tau}^{v} + \frac{\sum_{x^l \in KNBR_u^v} sim(TKF_u^v, TKF_x^l) \times (r_{x,d,\tau}^l - \bar{r}_{x,\tau}^l)}{\sum_{x^l \in KNBR_u^v} |sim(TKF_u^v, TKF_x^l)|}, \tag{15}$$

where $\bar{r}_{u,\tau}^v / \bar{r}_{x,\tau}^l$ is the topic rating of the target worker *u*/worker *x* for task *v*/*l*, derived from the worker's average rating of documents in the recommended topic τ; TKF_u^v / TKF_x^l is the topic-level KF of the target worker *u*/worker *x* for task *v*/task *l*; $r_{x,d,\tau}^l$ is the rating given by worker *x* for a document *d* belonging to the recommended topic τ in task *l*; and $sim(TKF_u^v, TKF_x^l)$ is the KF similarity of worker *u* and worker *x*, derived by Eq. (8). If the target worker *u* does not rate any documents in τ, then $\bar{r}_{u,\tau}^v$ is replaced by the average rating of all his/her documents.

To recommend task-related documents to a target worker, it is necessary to collect data with explicit ratings. Many recommender systems and recommendation methods use such ratings to represent users' preferences. Similarly, our recommendation methods use knowledge workers' document ratings to predict other documents that may be useful to a target worker's task, as shown in Eq. (15). Each knowledge worker gives explicit ratings to the documents referenced during the task's execution, while documents related to different tasks are re-rated by different workers. The ratings are used to gauge a worker's perceptions about the usefulness and relevance of documents for a specific task. The stronger the worker's perceptions of the usefulness or relevance of a document for the task at hand, the higher the rating he/she will give the document. Such ratings are subjective because they are based on the worker's perspective. Moreover, since a document may be referenced by different workers as they execute their specific tasks, it will be given different ratings based on how the workers perceive its usefulness and relevance to their tasks.

The sequential rules with high matching scores are selected to recommend topics. In other words, topics with high scores in the consequent part of a rule are recommended to the target worker at time *T*. The KSR method predicts ratings for documents that belong to the recommended topics and gives them a high priority for recommendation. Unlike traditional methods, KSR recommends documents to the target worker based on the selected sequential rules and the document ratings. Note that the KSR method does not consider the similarity of workers' preferences when calculating the predicted rating of a document.

5.3. The hybrid PCF–KSR method

The hybrid PCF–KSR recommendation method linearly combines the preference-similarity-based CF method (PCF) with the KSR method to recommend documents to a target worker, as shown in Fig. 4. The PCF method is the traditional CF method that makes recommendations according to workers' preferences for codified knowledge. To recommend a document, the neighbors of a target worker are selected based on the similarities of the workers' preference ratings. Pearson's correlation coefficient is used to find similar workers based on the document rating vectors. Then, PCF–KSR predicts the rating of a document by linearly combining the predicted ratings calculated by the two methods. One part of the rating is derived by the PCF method based on the document ratings and the preferences of the target worker's neighbors. The other part is derived by the KSR method described in Section 5.2. Because a worker's knowledge flow may change over time, the hybrid method considers the worker's preference for documents as well as topic changes in his/her KF to make recommendations adaptively.

The predicted rating of a document *d* for a worker *u* executing a task *v* is derived by combining the PCF and KSR methods, as defined in Eq. (16):

$$\hat{p}_{u,d}^{v} = \beta_{PCF-KSR} \times \left[\bar{r}_u^v + \frac{\sum_{x^l \in PNB_u^v} PSim(u^v, x^l) \times (r_{x,d}^l - \bar{r}_x^l)}{\sum_{x^l \in PNB_u^v} |PSim(u^v, x^l)|}\right] + (1 - \beta_{PCF-KSR}) \times \hat{p}_{u,v,d}^{KSR}, \tag{16}$$

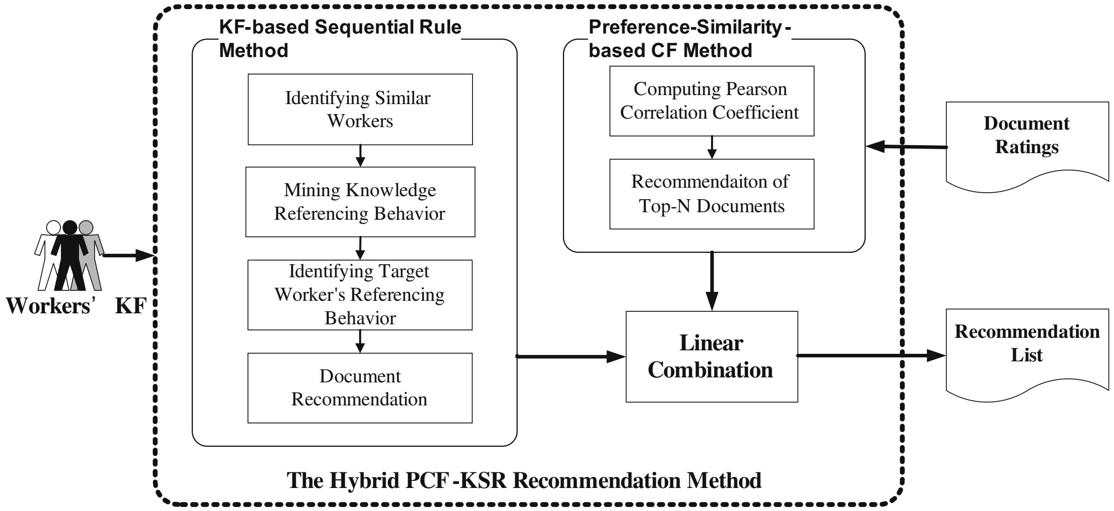
KF-based Sequential Rule Method
Identifying Similar Workers
Mining Knowledge Referencing Behavior
Identifying Target Worker's Referencing Behavior
Document Recommendation
Preference-Similarity-based CF Method
Computing Pearson Correlation Coefficient
Recommendaiton of Top-N Documents
Document Ratings
Workers' KF
Linear Combination
Recommendation List
The Hybrid PCF-KSR Recommendation Method

Fig. 4. The framework of the hybrid PCF–KSR method.

where $\bar{r}^{v}_{u}/\bar{r}^{l}_{x}$ is the average rating of documents for task v/task l given by the target worker u/worker x; $PSim(u^{v}, x^{l})$ is the similarity between the target worker u for task v and the neighbor worker x for task l, derived by Pearson's correlation coefficient; PNB^{v}_{u} is the set of neighbors of the target worker u for task v, selected by $PSim(u^{v}, x^{l})$; $r^{l}_{x,d}$ is the rating of a document d for task l given by worker x; $\hat{p}^{KSR}_{u,v,d}$ is the predicted rating of a document d for the target worker u engaged in task v based on the KSR method; and $\beta_{PCF-KSR}$ is the weighting used to adjust the relative importance of the PCF method and KSR method.

According to Eq. (16), a document in a recommended topic has a higher priority for recommendation than documents that are not in the recommended topics, based on their predicted ratings derived by the KSR method. Documents with high predicted ratings are used to compile a recommendation list, from which the top-N documents are chosen and recommended to the target worker.

5.4. The hybrid KCF–KSR method

The hybrid KCF–KSR method linearly combines the KF-similarity-based CF method (KCF) with the KSR method to recommend documents to a target worker, as shown in Fig. 5. The KCF method is based on the referencing behavior of neighbors with similar KFs, while the PCF method is based on the similarity of preference ratings derived by Pearson correlation coefficient. Like the PCF–KSR method, the predicted rating of a document is also derived by integrating two parts of the ratings. One part is obtained by the KCF method, while the other is obtained by the KSR method described in Section 5.2.

The hybrid KCF–KSR method predicts the rating of a document d for worker u engaged in task v by Eq. (17), and then determines which documents should be recommended.

$$\hat{p}^{v}_{u,d} = \beta_{KCF-KSR} \times \left[\bar{r}^{v}_{u} + \frac{\sum_{x^{l}\in KNB^{v}_{u}} sim(TKF^{v}_{u}, TKF^{l}_{x}) \times (r^{l}_{x,d} - \bar{r}^{l}_{x})}{\sum_{x^{l}\in KNB^{v}_{u}} |sim(TKF^{v}_{u}, TKF^{l}_{x})|} \right] + (1-\beta_{KCF-KSR}) \times \hat{p}^{KSR}_{u,v,d}, \tag{17}$$

where $\bar{r}^{v}_{u}/\bar{r}^{l}_{x}$ is the average rating of documents given by the target worker u/worker x engaged in task v/l; $r^{l}_{x,d}$ is the rating of a document d for task l given by worker x; TKF^{v}_{u}/TKF^{l}_{x} denotes the topic-level KF of the target worker u/worker x for task v/task l; $sim(TKF^{v}_{u}, TKF^{l}_{x})$ is the KF similarity of worker u and worker x, derived by Eq. (8); KNB^{v}_{u} is the set of neighbors of the target worker u for task v, selected according to their KF similarity scores; $\hat{p}^{KSR}_{u,v,d}$

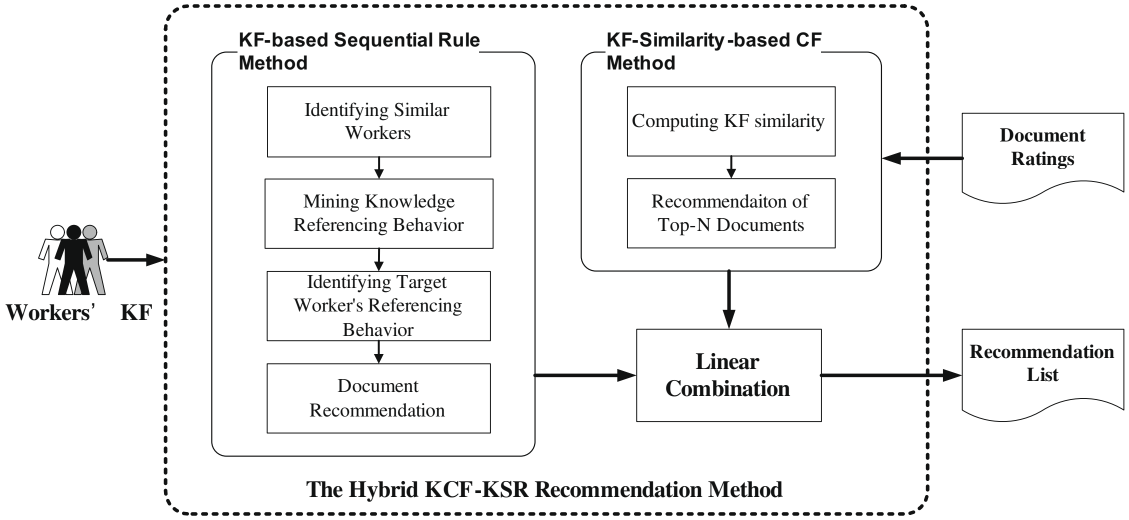
KF-based Sequential Rule Method
Identifying Similar Workers
Mining Knowledge Referencing Behavior
Identifying Target Worker's Referencing Behavior
Document Recommendation
KF-Similarity-based CF Method
Computing KF similarity
Recommendaiton of Top-N Documents
Document Ratings
Workers' KF
Linear Combination
Recommendation List
The Hybrid KCF-KSR Recommendation Method

Fig. 5. The framework of the hybrid KCF–KSR method.

is the predicted rating of a document d based on the KSR method; and $\beta_{KCF-KSR}$ is the weighting used to adjust the relative importance of the KCF method and the KSR method.

According to Eq. (17), a document in a recommended topic has a higher priority for recommendation than those documents that are not in the recommended topic. The KCF–KSR method considers the KF similarity of two workers, their preferences for documents, and topic sequences in the KF when making recommendations.

5.5. The hybrid ICF–KSR method

The hybrid ICF–KSR recommendation method linearly combines the item-based CF method (ICF) with the KSR method to recommend documents to a target worker, as shown in Fig. 6. The ICF method is the traditional item-based CF method (Sarwar et al., 2001) described in Section 2.6. The similar documents (neighbors) of a target document are selected based on the adjusted cosine similarities of the documents (Eq. (6)). Then, the predicted rating of the target document is computed by taking the weighted average of the target worker's ratings for similar documents (Eq. (5)).

The ICF method does not consider workers' referencing behavior when they perform tasks. To address this issue, we propose the hybrid ICF–KSR method, which integrates traditional item-based collaborative filtering and the KSR method to recommend documents that may meet workers' information needs. The ICF–KSR approach predicts the rating of a document by linearly combining the predicted ratings calculated by the two methods. One part of the rating is derived by the ICF method based on the target worker's ratings for documents similar to the target document. The other part is derived by the KSR method described in Section 5.2. A worker's knowledge flow may change over time. Thus, to make recommendations adaptively, the hybrid method considers documents similar to the target document, the worker's perceptions about the usefulness of the documents, and the topic sequences in his/her KF.

The hybrid ICF–KSR method predicts a rating for a document d for worker u performing a task v by using Eq. (18), and then determines the documents that should be recommended.

$$\hat{p}^{v}_{u,d} = \beta_{ICF-KSR} \times \left[\frac{\sum_{i\in I_d} ACSim(d,i) \times r^{v}_{u,i}}{\sum_{i\in I_d} |ACSim(d,i)|}\right] + (1-\beta_{ICF-KSR}) \times \hat{p}^{KSR}_{u,v,d}, \tag{18}$$

where $r^{v}_{u,i}$ is the rating of the usefulness of a document i given by worker u for task v; $ACSim(d,i)$ is the adjusted cosine similarity between document d and document i; I_d is the set of documents similar to document d, selected according to their adjusted cosine similarities; $\hat{p}^{KSR}_{u,v,d}$ is the predicted rating of document d for the target worker u engaged in task v based on the KSR method; and $\beta_{ICF-KSR}$ is the weighting used to adjust the relative importance of the ICF method and the KSR method. According to Eq. (18), a document in a recommended topic has a higher priority for recommendation than documents that are not in the recommended topic.

6. Experiments and evaluations

In this section, we conduct experiments to compare and evaluate the recommendation quality for the hybrid PCF–KSR, KCF–KSR and ICF–KSR methods, and then have some discussions about these experimental results. Next, we will describe the experiment setup in Section 6.1, discuss the experiment results and evaluations in Section 6.2, and have some discussions in Section 6.3.

6.1. Experiment setup

To demonstrate that knowledge flows can support the recommendation of task-relevant knowledge (documents) to knowledge workers, experiments were conducted on a dataset from a real application domain, namely, research tasks in the laboratory of a research institute. The dataset contained information about the access behavior of each knowledge worker engaged in performing a specific task, e.g., writing a research paper or conducting a research project. To accomplish their tasks, the workers needed various documents (research papers). Besides the documents, other information, such as when the documents were referenced and the document ratings, is necessary for implementing our methods. Since it is difficult to obtain such a dataset, using the real application domain restricts the sample size of the data in our experiments.

The dataset is based on the referencing behavior of 14 knowledge workers in a research laboratory and 424 research papers used to evaluate the proposed methods. Specifically, it contains information about the content of the documents, the times they were referenced, and the document ratings given by workers. For each worker, the documents and the times at which they were referenced are used to identify the worker's referencing behavior when performing a task.

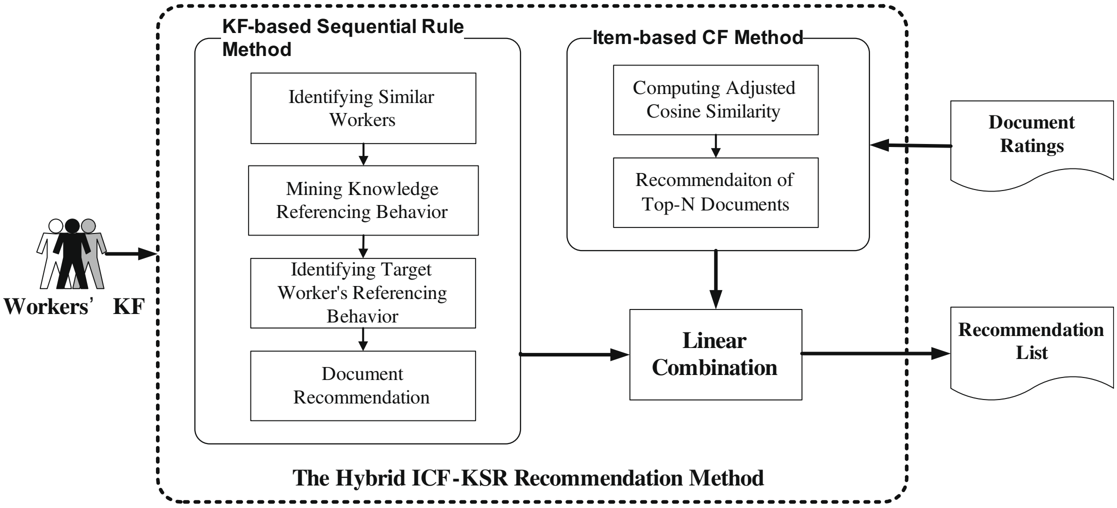
KF-based Sequential Rule Method
Identifying Similar Workers
Mining Knowledge Referencing Behavior
Identifying Target Worker's Referencing Behavior
Document Recommendation
Item-based CF Method
Computing Adjusted Cosine Similarity
Recommendaiton of Top-N Documents
Document Ratings
Workers' KF
Linear Combination
Recommendation List
The Hybrid ICF-KSR Recommendation Method

Fig. 6. The framework of the hybrid ICF–KSR method.

The document rating, which is given by a worker and on a scale of 1–5, indicates whether a document is perceived as useful and relevant to a task. A high rating, i.e., 4 or 5, indicates that the document is perceived as useful and relevant to the task at hand; while a low rating, i.e., 1 or 2, suggests that the document is deemed not useful. If a document has been referenced by a worker without being assigned a rating value, it is given a default rating of 3.

In our experiment, the dataset is divided according to the time order of the documents accessed by knowledge workers as follows: 70% for training and 30% for testing. The testing set contains documents with access time more close to the current time period. The training set is used to generate recommendation lists, while the test set is used to verify the quality of the recommendations. In the experiments, we evaluate and compare the performance of traditional CF methods and our KF-based recommendation methods, namely the hybrid PCF–KSR method, the hybrid KCF–KSR method, and the hybrid ICF–KSR method.

We use the Mean Absolute Error (MAE), which is widely used in recommender systems (Breese et al., 1998; Herlocker et al., 1999; Herlocker et al., 2004; Shardanand and Maes, 1995), to evaluate the quality of recommendations derived by our methods. MAE measures the average absolute deviation between a predicted rating and the user's true rating (Shardanand and Maes, 1995), as shown in Eq. (19).

$$MAE = \frac{\sum_{i \in Z, i=1}^{n} |p_i - q_i|}{n}, \tag{19}$$

where MAE is the mean absolute error; Z is the test set of a target worker, which consists of n predicted documents; p_i is the predicted rating of document i; and q_i is the real rating of document i. The lower the MAE, the more accurate the method will be. The advantages of this measurement are that its computation is simple and easy to understand and it has well studied statistical properties for testing the significance of a difference.

6.2. Experiment results

We conduct several experiments to measure the quality of recommendations derived by our methods. To generate topic-level KFs, the documents in the data set are grouped into clusters by the single-link hierarchical clustering method described in Section 4.1. To determine the threshold value that yields the best clustering result, we adjust the threshold value systematically in decrements of 0.05 ranging from 0.5 to 0.2 to generate different clustering results, each of which is evaluated by using the quality measure defined in Section 2.3.1. The cluster with the best quality measure generated by setting the threshold value at 0.3 is selected as our clustering result; it contains 8 clusters. Based on the clustering results, topic-level KFs are generated by mapping documents from the codified-level KFs into their corresponding clusters for each knowledge worker. Finally, by considering the topic-level and codified-level KFs, the hybrid PCF–KSR and KCF–KSR methods recommend task-related documents to users. In the following subsections, we discuss the experiment results.

6.2.1. Evaluation of the hybrid PCF–KSR method

In this experiment, we evaluate the performance of the hybrid PCF–KSR method. The parameters, α and $\beta_{PCF-KSR}$, may affect the quality of the recommendations; α is used to calculate the KF similarity (Eq. (8)), while $\beta_{PCF-KSR}$ is used to predict a document's rating. We set various values for these parameters and determine the settings that yield the best recommendation performance. The experiment was conducted by systematically adjusting the values of α in increments of 0.1, and the optimal value (i.e., the lowest MAE value) was chosen as the best setting. Based on the experiment results, we set α = 0.3 in all the following experiments.

We evaluate how the $\beta_{PCF-KSR}$ values and the number of neighbors, k, affect the recommendation quality, as shown in Fig. 7. The parameter $\beta_{PCF-KSR}$, whose value ranges from 0.1 to 1, represents the relative importance of the PCF method and KSR method in Eq. (16). The experiment was conducted using various numbers of neighbors (parameter k) to derive the predicted ratings. Fig. 7 shows that the lowest MAE value generally occurs when $\beta_{PCF-KSR}$ is 0.5.

Fig. 8 compares the hybrid PCF–KSR method with the traditional CF method (PCF method). The predicted rating of a document is derived in two parts by the PCF method and the KSR method respectively. The part derived by the PCF method is based on the document ratings of the target worker's neighbors, while the other part is derived by the KSR method based on documents in the recommended topics and sequential rules generated from the KFs of the target worker's neighbors. If a document is in the recommended topic, the KSR part of PCF–KSR can be used to adjust the predicted rating of the document. Therefore, the PCF–KSR method ensures that documents in the recommended topics have a high priority for recommendation to the target worker. In the experiment, we set α = 0.3 and $\beta_{PCF-KSR}$ = 0.5, and select the top-5 sequential rules with high rule matching scores. The experiment results show that the PCF–KSR method outperforms the traditional CF method (PCF method) under various numbers of neighbors (parameter k). That is, the KSR method improves the recommendation quality of the PCF method. In other words, the PCF–KSR method is effective in recommending documents to the target worker, and it improves on the quality of the recommendations derived by the PCF method alone.

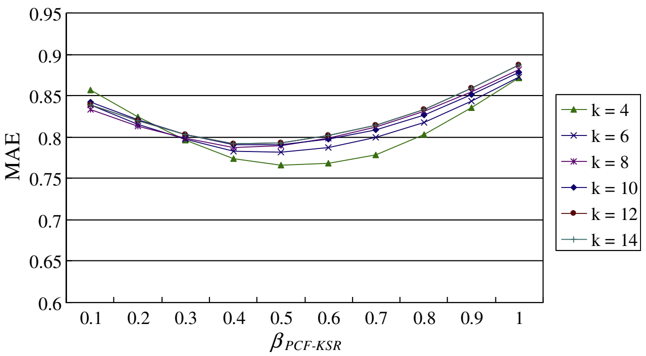

Fig. 7. The performance of the hybrid PCF–KSR method with various k and $\beta_{PCF-KSR}$ values.

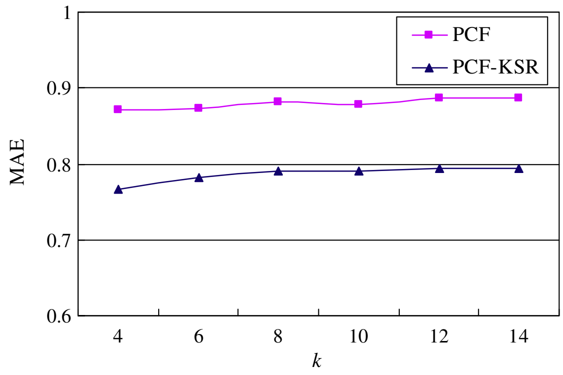

Fig. 8. Comparison of the hybrid PCF–KSR and PCF methods under different k.

6.2.2. Evaluation of the hybrid KCF–KSR method

Similar to the evaluation of the hybrid PCF–KSR method, we first determine the value of $\beta_{KCF-KSR}$ for the KCF–KSR method. The $\beta_{KCF-KSR}$ parameter, whose value ranges from 0.1 to 1, represents the relative importance of the KCF method and the KSR method. We set $\alpha = 0.3$ when calculating the KF similarity. The results show that the smallest value of MAE usually occurs when $\beta_{KCF-KSR} = 0.5$ for different the numbers of neighbors (k). Thus, in this experiment, $\beta_{KCF-KSR}$ is set at 0.5 for the KCF–KSR method.

To evaluate the performance of the KCF–KSR method, we compare it with the KF-similarity-based CF method (KCF) by setting $\beta_{KCF-KSR}$ at 1, as shown in Fig. 9. Note that when $\beta_{KCF-KSR} = 1$, the predicted rating of a document is derived totally by the KCF method, which only uses the document ratings of the target worker's neighbors with similar KFs to make recommendations. The experiment results demonstrate that the hybrid KCF–KSR outperforms the KCF method. In other words, considering workers' knowledge referencing behavior can enhance the quality of recommendations.

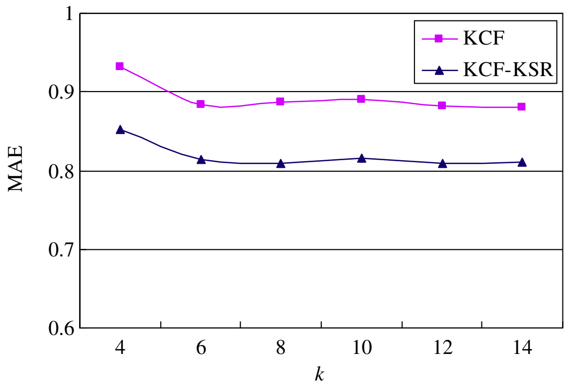

Fig. 9. Comparison of the hybrid KCF–KSR and KF methods under different k.

6.2.3. Evaluation of the hybrid ICF–KSR method

This experiment evaluates the performances of ICF and ICF–KSR methods. Once again we have to determine the value of the $\beta_{ICF-KSR}$ parameter in the range 0.1 to 1 to represent the relative weights of the ICF method and the KSR method. The results show that the smallest value of MAE usually occurs when $\beta_{ICF-KSR} = 0.4$ under various number of neighbors (k). Relatively, KSR is more important than ICF in the hybrid ICF–KSR method because the weight of KSR is higher than that of ICF. Thus, $\beta_{ICF-KSR}$ is set at 0.4 for the ICF–KSR method in this experiment.

To assess the impact of considering workers' referencing behavior on the ICF–KSR method, we compare it with the ICF method by setting $\beta_{ICF-KSR}$ at 1, as shown in Fig. 10. Setting $\beta_{KCF-KSR} = 1$ means that the predicted rating of a document is derived totally by the ICF method, which only utilizes the adjusted cosine similarity measures between documents to make recommendations. The hybrid ICF–KSR method takes this issue into account. Fig. 10 demonstrates that the hybrid ICF–KSR method performs better than the ICF method under various numbers of neighbors (parameter k). The experiment results show that considering workers' knowledge referencing behavior under the KSR method improves the recommendation quality of the ICF method.

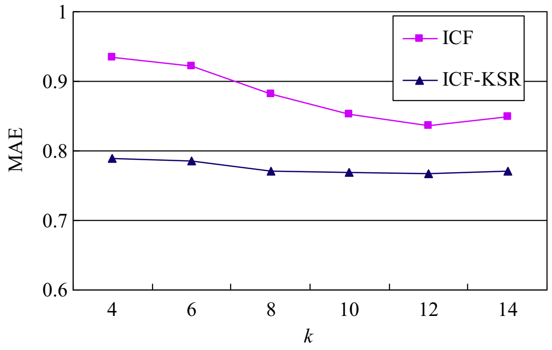

Fig. 10. Comparison of the hybrid ICF–KSR and KF methods under different k.

6.2.4. Comparison of all methods

To evaluate the recommendation performances of the different methods, we compare the three individual methods (the PCF, KCF and ICF methods) and the three hybrid methods (the PCF–KSR, KCF–KSR and ICF–KSR methods), as shown in Fig. 11.

When the number of neighbors, k, is less than 8, the PCF method yields the lowest MAE values, while the ICF method yields the highest values. However, when the value of k is more than 8, the ICF method outperforms the KCF and PCF methods. The recommendation performances of the PCF method and the KCF methods are very close.

In this experiment, we also compare the hybrid PCF–KSR, the hybrid KCF–KSR and the hybrid ICF–KSR methods, under various k (the number of neighbors). To obtain the MAE values of these methods, we set $\alpha = 0.3$, $\beta_{PCF-KSR} = 0.5$, $\beta_{KCF-KSR} = 0.5$ and $\beta_{ICF-KSR} = 0.4$. The results show that the hybrid ICF–KSR method generally outperforms the PCF–KSR and KCF–KSR methods, while the PCF–KSR method performs better than the KCF–KSR method.

To examine the differences between the KF-based methods and the traditional CF method, we performed a statistical hypothesis test, the paired t-test, under various k. The results show that the differences are statistically significant at the 0.01 level. Here, we only report the results of the t-test under $k = 8$. The mean, standard deviation (SD), and p-value of MAE for each pair of recommendation methods are listed in Table 1. The proposed hybrid methods, i.e., PCF–KSR, KCF–KSR and ICF–KSR, have smaller mean and generally smaller standard deviation scores than their individual methods. In terms of the p-value, the differences between the proposed hybrid methods and the individual CF-based methods are statistically significant.

From the above results, it is clear that the hybrid methods perform better than their individual methods. That is, the hybrid PCF–KSR, KCF–KSR and ICF–KSR methods perform better than PCF, KCF and ICF methods alone. The results show that the KF-based approaches can enhance the recommendation quality of traditional CF methods.

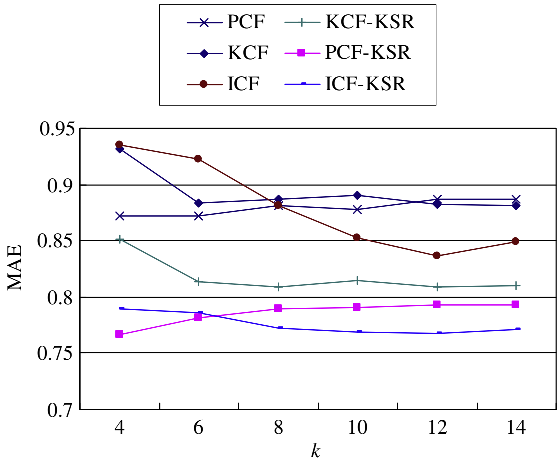

Fig. 11. The performances of the compared methods under different k.

Table 1
The *t*-test results for various recommendation methods with $k = 8$.

Recommendation method	Mean	*SD*	*t*-Test
PCF–KSR	0.7898	0.7189	$p = 0.0006$ (<0.01)
PCF	0.8814	0.7244	
KCF–KSR	0.8086	0.7581	$p = 0.0006$ (<0.01)
KCF	0.8865	0.7836	
ICF–KSR	0.7718	0.6880	$p = 0.0045$ (<0.01)
ICF	0.8814	0.6829	

6.3. Discussion

The experiment results demonstrate that the proposed KF-based hybrid methods, i.e., the PCF–KSR, KCF–KSR and ICF–KSR methods, improve the quality of document recommendation and outperform traditional CF methods. The three hybrid methods also perform better than the individual methods, i.e., the PCF, KCF, and ICF methods. However, the current study has some limitations. First, our experiments were conducted using a real application domain, i.e., research tasks in a research institute's laboratory. The domain restricted the sample size of the data and the number of participants in the experiments, since it is difficult to obtain a dataset that contains information that can be used for knowledge flow mining. Because of this limitation, in our future work, we will evaluate the proposed approach on other application domains involving larger numbers of workers, tasks and documents. Second, our evaluation focused on verifying the effectiveness of the proposed approach for recommending codified knowledge (documents) based on knowledge flows, rather than on user satisfaction or the system's usability. A study of user satisfaction or usability would add further insights into our system's ability to recommend task-relevant knowledge. In addition, the ratings given by people with different roles (e.g., professors and students) may have different influences on the recommendations. For example, it could be assumed that the rating given by a professor is more trustworthy than that given by a student. We will consider this issue in our future work.

7. Conclusions and future work

Knowledge is both abstract and dynamic. A worker's knowledge flow (KF) comprises a great deal of working knowledge that is difficult to acquire from an organizational knowledge base. In this paper, we have considered how to identify the knowledge flow of knowledge workers, and how to provide knowledge support based on KFs effectively. To the best of our knowledge, no existing approach focuses on providing relevant knowledge proactively based on KFs.

We propose KF-based recommendation methods, namely hybrid PCF–KSR, KCF–KSR and ICF–KSR methods, to proactively recommend codified knowledge for knowledge workers and enhance the quality of recommendations. These methods use KF-based sequential rule (KSR) method to recommend topics by considering workers' knowledge referencing behavior; and then adjust the predicted rating of documents belonging to the recommended topic. Moreover, they consider workers' preferences for codified knowledge, as well as their knowledge referencing behavior to predict topics of interest and recommend task-related knowledge. The collaborative filtering (CF) method, which is widely used to predict a target worker's preferences based on the opinions of similar workers, only considers workers' preferences for codified knowledge, but it neglects workers' referencing behavior for knowledge.

In the experiments, we evaluate the quality of recommendations derived by the proposed methods under various parameters and compare it with that of the traditional user-based/item-based CF method. The experiment results show that the proposed methods improve the quality of document recommendation and outperform the traditional CF methods. Additionally, using KF mining and sequential rule mining techniques enhances the performance of recommendation methods and increases the accuracy of recommendations. The KF-based recommendation methods provide knowledge support adaptively based on the referencing behavior of workers with similar KFs, and also facilitate knowledge sharing among such workers.

In our current work, a KF is simply regarded as a set of topics/codified knowledge objects arranged in a time sequence. However, a KF may have a complicated order structure with AND/OR, JOIN and SPLIT operations. In our future work, we will investigate a complex KF mining technique to model workers' KFs with an order structure that includes such operations. Moreover, the discovered topic is regarded as an abstraction of topic-related documents. Auto-summarization techniques (Radev et al., 2004; Salton et al., 1997) can be applied to extract the theme of a topic by summarizing the documents' contents. In a future work, we will investigate the use of such techniques to derive knowledge flows based on theme information. In addition, the domain restricted the sample size of the data and the number of participants in the experiments, since it is difficult to obtain a dataset that contains information that can be used for knowledge flow mining. We will evaluate the proposed approach on other application domains involving larger numbers of workers, tasks and documents. Moreover, the method of generating topic subsequences for identifying the target worker's knowledge referencing behavior is computationally expensive, especially for the large datasets. A more efficient method will be investigated in the future. Furthermore, we will study a group-based KF mining method to identify the KFs of groups of workers. Such groups may be interest groups or communities, where the workers have very similar KFs. A group may comprise many workers with similar KFs, and a worker may join many groups simultaneously according to his/her information needs. Based on the KF of a group, we will design recommendation techniques to share knowledge adaptively and effectively among workers in different groups or communities.

Acknowledgement

This research was supported by the National Science Council of the Taiwan under the grant NSC 96-2416-H-009-007-MY3.

References

Abecker, A., Bernardi, A., Hinkelmann, K., Kuhn, O., Sintek, M., 2000a. Context-aware, proactive delivery of task-specific information: the KnowMore project. Information Systems Frontiers 2 (3), 253–276.

Abecker, A., Bernardi, A., Maus, H., Sintek, M., Wenzel, C., 2000b. Information supply for business processes: coupling workflow with document analysis and information retrieval. Knowledge-Based Systems 13, 271–284.

Agrawal, R., Imielinski, T., Swami, A., 1993. Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD international conference on Management of data, Washington, D.C., United States, pp. 207–216.

Agrawal, R., Srikant, R., 1994. Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487–499.

Agrawal, R., Srikant, R., 1995. Mining sequential patterns. In: Proceedings of the Eleventh International Conference on Data Engineering, pp. 3–14.

Anjewierden, A., de Hoog, R., Brussee, R., Efimova, L., 2005. Detecting knowledge flows in Weblogs. In: 13th International Conference on Conceptual Structures (ICCS 2005), pp. 1–12.

Augusto, C., Maria Grazia, F., Silvano, P., 1995. Knowledge-based document retrieval in office environments: the Kabiria system. ACM Transactions on Information Systems 13 (3), 237–268.

Baeza-Yates, R., Ribeiro-Neto, B., 1999. Modern Information Retrieval. Addison-Wesley, Boston.

Breese, J.S., Heckerman, D., Kadie, C., 1998. Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, pp. 43–52.

Brown, J.S., Duguid, P., 2002. The Social Life of Information. Harvard Business School Press, Boston, MA, USA.

Charter, K., Schaeffer, J., Szafron, D., 2000. Sequence alignment using FastLSA. In: International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS, 2000), pp. 239–245.

Cho, Y.B., Cho, Y.H., Kim, S.H., 2005. Mining changes in customer buying behavior for collaborative recommendations. Expert Systems with Applications 28 (2), 359–369.

Chuang, S.L., Chien, L.F., 2004. A practical web-based approach to generating topic hierarchy for text segments. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management (CIKM), Washington, D.C., USA, pp. 127–136.

Dragunov, A.N., Dietterich, T.G., Johnsrude, K., McLaughlin, M., Li, L., Herlocker, J.L., 2005. TaskTracer: a desktop environment to support multi-tasking knowledge workers. In: Proceedings of the 10th International Conference on Intelligent User Interfaces, San Diego, California, USA, pp. 75–82.

Dubes, R.C., Jain, A.K., 1988. Algorithms for Clustering Data. Prentice-Hall, Inc.

Feldman, R., Sanger, J., 2007. The Text Mining Handbook: Advanced Approaches to Analyzing Unstructured Data, vol. 34. Cambridge University Press, New York, USA.

Glance, N., Arregui, D., Dardenne, M., 1998. Knowledge pump: community-centered collaborative filtering. In: Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering, pp. 83–88.

Hay, B., Wets, G., Vanhoof, K., 2001. Clustering navigation patterns on a website using a sequence alignment method. In: Proceedings of the17th International Joint Conference on Artificial Intelligence (IJCAI), Seattle, Washington, pp. 1–6.

Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J., 1999. An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, California, United States, pp. 230–237.

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T., 2004. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22 (1), 5–53.

Holz, H., Maus, H., Bernardi, A., Rostanin, O., 2005. A lightweight approach for proactive, task-specific information delivery. In: Proceedings of the 5th International Conference on Knowledge Management (I-Know), pp. 101–127.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: a review. ACM Computing Surveys (CSUR) 31 (3), 264–323.

Johnson, S.C., 1967. Hierarchical clustering schemes. Psychometrika 32 (3), 241–254.

Kaufman, L., Rousseeuw, P.J., 1990. Finding groups in data. An introduction to cluster analysis. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics. Wiley, New York.

Kim, S., Hwang, H., Suh, E., 2003. A process-based approach to knowledge-flow analysis: a case study of a manufacturing firm. Knowledge and Process Management 10 (4), 260–276.

Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J., 1997. GroupLens: applying collaborative filtering to Usenet news. Communications of the ACM 40 (3), 77–87.

Kruskal, J.B., 1983. An overview of sequence comparison: time warps, string edits, and macromolecules. SIAM Review 25 (2), 201–237.

Linden, G., Smith, B., York, J., 2003. Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing 7 (1), 76–80.

Liu, D.-R., Lai, C.-H., Huang, C.-W., 2008. Document recommendation for knowledge sharing in personal folder environments. Journal of Systems and Software 81 (8), 1377–1388.

Liu, D.-R., Shih, Y.-Y., 2005. Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences. Journal of Systems and Software 77 (2), 181–191.

Liu, D.-R., Wu, I.-C., Yang, K.-S., 2005. Task-based K-Support system: disseminating and sharing task-relevant knowledge. Expert Systems With Applications 29 (2), 408–423.

Mannila, H., Ronkainen, P., 1997. Similarity of event sequences. In: Proceedings of the Fourth International Workshop on Temporal Representation and Reasoning, Florida, USA, pp. 136–139.

Nonaka, I., Takeuchi, H., 1995. The Knowledge-creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press.

Oguducu, S.G., Ozsu, M.T., 2006. Incremental click-stream tree model: learning from new users for web page prediction. Distributed and Parallel Databases 19 (1), 5–27.

Polanyi, M., 1966. The Tacit Dimension. Doubleday, New York.

Radev, D.R., Jing, H., Stys, M., Tam, D., 2004. Centroid-based summarization of multiple documents. Information Processing and Management 40 (6), 919–938.

Rodriguez, O.M., Martinez, A.I., Favela, J., Vizcaino, A., Piattini, M., 2004. Understanding and supporting knowledge flows in a community of software developers. In: International Workshop on Groupware (CRIWG).

Rucker, J., Polanco, M.J., 1997. Siteseer: personalized navigation for the web. Communications of the ACM 40 (3), 73–76.

Salton, G., Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Information Processing and Management 24 (5), 513–523.

Salton, G., Singhal, A., Mitra, M., Buckley, C., 1997. Automatic text structuring and summarization. Information Processing and Management 33 (2), 193–207.

Sarwar, B., Karypis, G., Konstan, J., Reidl, J., 2001. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, Hong Kong, pp. 285–295.

Shardanand, U., Maes, P., 1995. Social information filtering: algorithms for automating, word of mouth. In: Proceedings of the SIGCHI conference on Human factors in computing systems (CHI, 95), Denver, Colorado, United States, pp. 210–217.

Van RijsBergen, C.J., 1979. Information Retrieval. Butterworths, London.

Wu, I.C., Liu, D.R., Chen, W.H., 2005. Task-stage knowledge support: coupling user information needs with stage identification. In: IEEE International Conference on Information Reuse and Integration (IRI, 2005), pp. 19–24.

Yun, H., Ha, D., Hwang, B., Ho Ryu, K., 2003. Mining association rules on significant rare data using relative support. The Journal of Systems and Software 67 (3), 181–191.

Zhao, Y., Karypis, G., Fayyad, U., 2005. Hierarchical clustering algorithms for document datasets. Data Mining and Knowledge Discovery 10 (2), 141–168.

Zhuge, H., 2002. A knowledge flow model for peer-to-peer team knowledge sharing and management. Expert Systems with Applications 23 (1), 23–30.

Zhuge, H., 2006a. Discovery of knowledge flow in science. Communications of the ACM 49 (5), 101–107.

Zhuge, H., 2006b. Knowledge flow network planning and simulation. Decision Support Systems 42 (2), 571–592.

Zhuge, H., Guo, W., 2007. Virtual knowledge service market – for effective knowledge flow within knowledge grid. Journal of Systems and Software 80 (11), 1833–1842.

Chin-Hui Lai is a PhD student of the Institute of Information Management, National Chiao Tung University. She received her B.S. and M.S. degrees in Department of Information Management from the National Taiwan University of Science; Technology and Institute of Information Management from National Chiao Tung University, Taiwan, in 2001 and 2004, respectively. Her research interests include recommender systems, knowledge management and electronic commerce.

Duen-Ren Liu is a professor of the Institute of Information Management, National Chiao Tung University, Taiwan. He received the B.S. and M.S. degrees in Computer Science and Information Engineering from the National Taiwan University, Taiwan, in 1985 and 1987, respectively. He received the Ph.D. degree in Computer Science from the University of Minnesota in 1995. His research interests include information systems, recommender systems, electronic commerce, workflow systems and knowledge management.