Expert Systems with Applications 41 (2014) 4000–4009

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A framework for tag-aware recommender systems

Hyunwoo Kim *, Hyoung-Joo Kim

School of Computer Science and Engineering, Seoul National University, Seoul 151-742, Republic of Korea

ARTICLE INFO

Keywords:
Recommendation
Social tagging system
Tags
Hybrid framework

ABSTRACT

In social tagging system, a user annotates a tag to an item. The tagging information is utilized in recommendation process. In this paper, we propose a hybrid item recommendation method to mitigate limitations of existing approaches and propose a recommendation framework for social tagging systems. The proposed framework consists of tag and item recommendations. Tag recommendation helps users annotate tags and enriches the dataset of a social tagging system. Item recommendation utilizes tags to recommend relevant items to users. We investigate association rule, bigram, tag expansion, and implicit trust relationship for providing tag and item recommendations on the framework. The experimental results show that the proposed hybrid item recommendation method generates more appropriate items than existing research studies on a real-world social tagging dataset.

1. Introduction

A recommendation is a task in which user interests are identified and items of high relevance are recommended based on user preferences. There has been growing interest in recommender systems as a way to deal with information overload (Adomavicius & Tuzhilin, 2005). Information overload is a situation in which a user cannot make a decision or search through all items in a system because the amount of data in the system is too large. Recommender systems help users mitigate this situation by providing relevant items. A recommender system utilizes an item's content information and user ratings. By analyzing user items and their metadata, the recommender system searches for similar items and recommends those relevant to the user. By analyzing the user's rating patterns and preferences, the recommender system searches for similar users, and recommends items rated highly by similar users.

Many researchers have recently investigated recommender systems and achieved significant results. Collaborative filtering is based on user similarity, and it is one of the most successful approaches in recommender systems. However, applying collaborative filtering to recommender systems encounters three problems: sparsity, cold-start problem, and scalability issues (Adomavicius & Tuzhilin, 2005; Lee, Yang, & Park, 2004). A sparse *user-item* matrix causes a sparsity problem. Sparsity is a problem in which the number of items in a system is so large that even the most active users cannot rate all of its items, and can only rate a small subset of items. New users and items in the system cause cold-start user and item problems. A new user in a recommender system does not provide enough user interest information, preventing the system from providing appropriate recommendations. In addition, a new item may not be recommended to users if no one has yet rated the item. To generate recommendable items, the system conducts a large amount of computations. As the amount of data in the system increases, a scalability problem arises.

The same situation occurs in social tagging systems. In a social tagging system, users annotate tags to an item. A tag is an annotated keyword of an item, such as a bookmark, movie, photo, or user. One of the main purposes of social tagging is retrieval (Ames & Naaman, 2007). A tag describes an item, acting as its metadata, or expresses a user's impression of the item. For example, users annotate tags such as *good*, *bad*, or *cool* to the item. Through social tagging, users form a folksonomy and create non-hierarchical categories or indexes for retrieval. Social tagging has been essential to the success of Websites, such as Flickr, Delicious, and YouTube (Das, Thirumuruganathan, Amer-Yahia, Das, & Yu, 2012).

Tag and item recommendations are two recommendation problems inherent to social tagging systems. Tag recommendation supports a user tagging process. The system recommends relevant tags of an item to the users. When a user tries to annotate an item, the system provides appropriate tags for the item. Through a tag recommendation process, the system helps users annotate correct and unambiguous tags, and enriches the tagging information through recommendations. For an item recommendation, the system analyzes user profiles and recommends items related to user interests. Tag-aware recommender systems utilize the tagging information of the users. This additional information can be utilized by the recommender system. Tagging data forms a ternary relation between users, items and tags where typical data forms a binary relation between users and items (Rendle, Balby Marinho,

* Corresponding author. Tel.: +82 28801830.
E-mail address: hwkim@idb.snu.ac.kr (H. Kim).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.12.019

Nanopoulos, & Schmidt-Thieme, 2009). By utilizing tagging information, recommender systems can implicitly extract users' similar interests and similar points of view on items because of the difference of scalar and textual values between ratings and tags (Zhao et al., 2008).

Generally, content-based method have over-specification problem and collaborative filtering method have cold-start problem. The performance of the hybrid approach is better than each individual approach (Adomavicius & Tuzhilin, 2005) because it alleviates the drawbacks of both the content-based and collaborative filtering methods. In this paper, we integrate previous research studies into a hybrid item recommendation method to alleviate the limitations of weak recommenders. In our previous research studies on item recommendation, we investigated two different approaches: a content-based approach and a collaborative filtering approach. In the content-based approach, we expanded a cold-start user's tag set and utilized temporal information of a social tagging system to generate relevant items (Kim & Kim, 2012b). In the collaborative filtering approach, we extracted an implicit trust relationship between users based on the tagging information (Kim & Kim, 2012a). A trust relationship is different from user similarity, which is utilized through conventional collaborative filtering methods. User similarity is symmetric and has no direction. In contrast, a trust relationship is asymmetric and has directions. In a user similarity metric, if user u_a is analogous to user u_b, user u_b is also analogous to user u_a. In a trust relationship, if user u_a trusts user u_b, user u_b might or might not trust user u_a. There is no dependency of the trust relationship between two users. We utilize linear combination of the content-based and collaborative filtering approaches to mitigate limitations of two approaches.

We also propose a framework for tag-aware recommender systems to obtain the benefit from integrating tag and item recommendation in a single system. The proposed framework provide tag and item recommendations in a single system. Tag recommendation augments the size of the tagging data and enriches the quality of the tagging information. At the same time, high quality tagging information enables more precise recommendations to be provided.

The remainder of this paper is as follows. Section 2 reviews some existing research studies on tag and item recommendations in social tagging systems. Section 3 describes the proposed framework. Section 4 represents an experimental evaluation of the proposed hybrid approach for item recommendation. Finally, in Section 5, we summarize our proposal and present directions for future work.

2. Related work

In this section, we present research studies on tag and item recommendations for social tagging systems.

2.1. Tag recommendation

One of the first research studies on tag recommendation (Xu, Fu, Mao, & Su, 2006) investigated the tag co-occurrence frequency. The co-occurrence frequency between two tags is measured based on the number of times that two tags are annotated to the same item. If two tags are closely related, their co-occurrence frequency will be high. In contrast, if two tags are not related at all, their co-occurrence frequency will be very low. Sigurbjörnsson and Van Zwol (2008) also utilize the tag co-occurrence frequency in their recommendation tags. The tag co-occurrence frequency is calculated on either the whole dataset or the user's personomy. Personomy is a user folksonomy, and is a non-hierarchical taxonomy formed through social tagging. The tag co-occurrence frequency in a folksonomy is reflected by the general meaning of two tags, and the tag co-occurrence frequency in the personomy is reflected by the user's personal use of two tags in the social tagging system. The relatedness of two tags will differ under each situation. When the system fails to understand the correct meaning of a tag, the recommendation performance decreases because the tagging information incurs synonym, polysemy, and level variation problems (Golder & Huberman, 2006).

Wu et al. proposed a multi-modality recommendation to mitigate the weakness of tag co-occurrence frequency (Wu, Yang, Yu, & Hua, 2009). They formulated a tag recommendation as a learning problem. Tag-content correlation and image conditioned tag correlation are additionally combined to rank the tags. By building a visual language mode (VLM), the system correctly disambiguates the meanings of social tags.

Most existing research studies focused on the accuracy of the recommendation but Song et al. considered the efficiency issue (Song et al., 2008). They proposed a highly-automated novel framework for real-time tag recommendation. Utilizing Spectral Recursive Embedding and two-way Poisson Mixture Model, the proposed framework recommends tags efficiently and effectively on real-world large-scale tagging datasets. Song et al. also proposed a prototype-based method for tag recommendation (Song, Zhang, & Giles, 2011). The proposed approach searches for the most informative prototypes and recommends tags to a document utilizing multilabel classification. Their two methods are document-centered tag recommendation approaches. In practice, the methods are more robust than user-centered approaches due to the rich information in the documents.

The basic building block of social tagging is the triple $\langle$*user*, *item*, *tag*$\rangle$. When user u annotates item i using tag t, the triple $\langle u, i, t\rangle$ is stored in the system's dataset. Tagging information has a ternary relationship, but most recommender systems split the ternary relationship into three binary relationships: user-tag, user-item, and item-tag relationships. To generate recommendable tags, the system analyzes binary relationships. However, the process of splitting a ternary relationship creates a loss of information. To avoid such loss of information and preserve the ternary relationship of social tagging information, Symeonidis et al. conduct tensor dimensionality reduction (Symeonidis, Nanopoulos, & Manolopoulos, 2008).

Tag recommendation using personomy or personal information shows a better performance (Lipczak, 2008). When a user annotates a bookmark, the system analyzes documents on the user's desktop computer (Chirita, Costache, Nejdl, & Handschuh, 2007). Static tag recommendation methods recommend a list of tags, and the user then selects their preferred tags. For dynamic tag recommendation methods, a list of relevant tags is generated whenever the user selects a tag. Garg and Weber propose a method that not only recommends related tags but also dynamically updates the results when every additional tag is selected (Garg & Weber, 2008).

2.2. Item recommendation

Item recommendation has become a favored topic since the first appearance of research papers on collaborative filtering (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994). Recommendation methods can be classified into three categories, content-based methods, collaborative methods, and hybrid methods (Adomavicius & Tuzhilin, 2005), and recommendation methods for social tagging systems can be similarly categorized (Milicevic, Nanopoulos, & Ivanovic, 2010). Content-based methods analyze the item content itself, or the metadata of the item. The system searches for similar items by analyzing the content. The system recommends items similar to a user's previously rated items. Collaborative filtering analyzes a user's preference, and the system searches for similar users whose preference is analogous to the target user.

The system recommends items that are highly rated by similar users. Hybrid methods combine content-based and collaborative methods. Hybrid methods mitigate certain limitations of each method and provide results that are more appropriate.

In social tagging systems, tagging information is one of the most significant factors used in a recommendation. Tagging information is useful for multimedia content because such content contains little textual information. Tagging information is an additional bridge between users, between users and items, or between items. Recommendation methods used in social tagging systems can be classified into three categories. The first type of approach is analogous to conventional recommender system methods with tagging information (Guy, Zwerdling, Ronen, Carmel, & Uziel, 2010; Han, Cai, Shao, & Li, 2012; Peng, Zeng, Zhao, & Wang, 2010; Zhao et al., 2008). The second type of approach utilizes tagging information for the learning models (Guan et al., 2010; Shepitsen, Gemmell, Mobasher, & Burke, 2008). Finally, the third type of approach utilizes tagging information for statistical calculations (Cai, Zhang, Luo, Ding, & Chakravarthy, 2011; Symeonidis, Nanopoulos, & Manolopoulos, 2010).

Guy et al. utilize a company's social network and tags for item recommendation (Guy et al., 2010). They apply a collaborative filtering method using this social network. They extract similarity relationships from common tags, common items, and accompanying comments. Peng et al. (2010) utilize tagging information for a collaborative filtering method, and Han et al. (2012) utilize tag co-occurrence frequency to improve their recommendations. Zhao et al. utilize collaborative tagging to improve recommendation (Zhao et al., 2008). They calculate the semantic similarity between two tags using the tags' path length in WordNet graph. Hierarchical clustering used in machine learning improves personalized recommendations in a social tagging system (Shepitsen et al., 2008). Graph-based subspace learning has also been investigated to recommend documents in social tagging services (Guan et al., 2010). Users, tags, and documents are represented in the same space. In the graph, the connectivity structure of the source graph is preserved so that if two objects are strongly connected, two related objects are placed close to each other. Tagging information cannot be represented in a matrix because it has an inherent ternary relationship. Cai et al. (2011) and Symeonidis et al. (2010) utilize the tensor of tripartite relationships of tagging information. They indicate tagging information as a tensor and improve the recommendation quality. Karatzoglou, Amatriain, Baltrunas, and Oliver (2010) also utilize n-dimensional tensor factorization for multiverse recommendation.

3. Tag-aware recommender system

Tagging information is useful for recommender systems because the motivation for tagging is later retrieval. The purpose of tagging fits the purpose of the recommendation. Collaborative filtering fails in diverse or mixed domains (Herlocker, Konstan, Terveen, & Riedl, 2004) but tagging information generates a high-quality recommendation in heterogeneous domains. For instance, if an annotated tag is *Michael Jackson*, the recommender system may suggest music, news articles, books, or movies about Michael Jackson. In social tagging systems, recommendation method should therefore consider tagging information in the process. If two users have commonly tagged items, it implies that the two users have similar preferences and interests. If two users have common tags for the same item, it implies that the two users have a similar opinion of the items. Leveraging the tagging information makes the tag metadata of the items, and is suitable for multi-domain recommender systems, alleviates a lack of item descriptions, and integrates various ranking semantics.

In a tag-aware recommender system, there are two recommendation problems: tag recommendation and item recommendation. We propose an integrated framework for tag-aware recommender systems. Tag recommendation enriches the tagging information and improves the user experience. Item recommendation based on tagging information provides recommendation results that are more accurate than those based on the rating information.

Fig. 1 illustrates the overall framework of the proposed approach. When a user annotates an item using a tag input model, the tag recommendation engine recommends tags to the user and stores the selected results in a database. The tag recommendation engine utilizes the association rule and bigram to generate recommendable tags. Association rule alleviates homonym problem when we utilize bigram approach. When the system recommends items to the user through the recommendation module, the item recommendation engine recommends items to the user and stores the selected results in the database. The item recommendation engine utilizes tag expansion, temporal information, and trust relationships. Tag expansion that is content-based approach alleviate new user and item problems when we utilize trust relationships. Each recommendation engine interacts with the learning module, and the learning module interacts with the database. The learning module obtains relevant data from database to learn recommendation models and store resultant data, such as association rule, bigram tag co-occurrence frequency, divergence value, and probability mass function value. Item and tag recommendation engines utilize the data to generate recommendable objects. We elucidate the details of each recommendation method in Sections 3.1 and 3.2, respectively.

3.1. Tag Recommendation

While users may be aware of the benefits of tagging, the number of users who annotate tags is relatively small. Users may find annotating tags an annoying task and may not recognize which tags should be annotated to obtain better retrieval result. A tag recommendation is a process that helps users annotate tags. When a user annotates an item, the system recommends appropriate tags to the item. This process alleviates the ambiguity problem of tags, improves the user experience, and increases the quality of the tags.

In tag recommendation, we previously proposed a method using association rule and bigram approach (Kim, Lee, Shin, & Kim, 2009). We have adopted this approach into the proposed framework. This tag recommendation approach utilizes the collective knowledge of users. When a user annotates tags to an item, the system recommends relevant tags using both tags entered by the user and tagging information from the dataset. In this approach, three score metrics are used: A_{score}, B_{score}, and T_{score}. A represents an association rule, B represents bigram approach, and T represents the tagging information. We describe each score in Sections 3.1.1, 3.1.2, and 3.1.3, respectively.

3.1.1. A_{score}

An association rule is a method for extracting the relatedness of various items in data mining research. An association rule is extracted from the tagging dataset. In this approach, we utilize an association rule to obtain the relationship between relevant tags. An association rule can avoid any ambiguity problems in a tag recommendation. As mentioned in Section 2.1, most tag recommendation approaches consider the tag co-occurrence frequency. However, based on the tag co-occurrence frequency, a tag sense disambiguation problem occurs (Lee, Kim, Shin, & Kim, 2009). One major meaning of a tag dominates other meanings of the tag. If a tag has different meanings, the tag co-occurrence frequency is unable to allow various meanings to be ascertained. For instance, if the tags *apple* and *farm* are entered, a system based

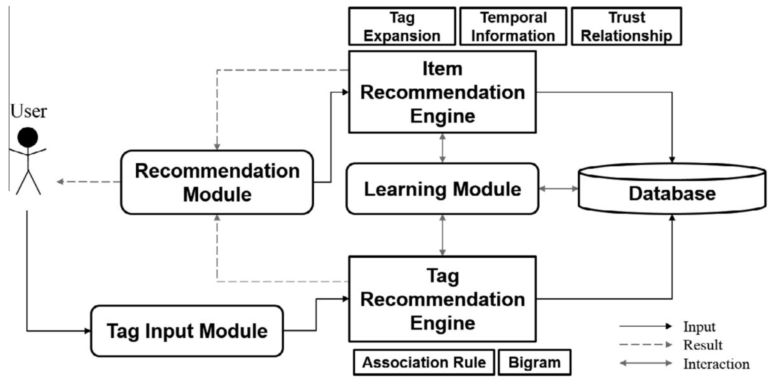

Fig. 1. Overall framework of the proposed approach.

on the tag co-occurrence frequency may recommend *iPhone* and *fruit* because the tag co-occurrence frequency of *iPhone* and *fruit* with the tag *apple* is high. However, when *apple* and *farm* are annotated together, the tag *apple* does not indicate the name of a company. An association rule considers each tag at the same time to avoid this problem. In marketing analysis, an association rule *beer*, *water* → *diaper* indicates that many customers who purchase beer and water are inclined to purchase diapers. In a tag recommendation, an association rule *iPhone*, *Steve Jobs* → *Apple* indicates that many users who annotate *iPhone* and *Steve Jobs* are also inclined to annotate *Apple*.

A_{score} is a confidence value of an association rule. A_{score} of association rule r can be defined as follows:

$$A_{score}(r) = \text{confidence}(\text{r})$$

A confidence value is a conditional probability of an association rule. The confidence value of association rule r, *iPhone*, *Steve Jobs* → *Apple*, can be defined as follows:

$$\text{Confidence}(\text{r}) = \frac{P(iPhone, Steve\ Jobs, Apple)}{P(iPhone, Steve\ Jobs)}$$

3.1.2. B_{score}

A bigram approach is a kind of n-gram approach in natural language processing. Only adjacent words are considered in a bigram approach. People think through associations. They have one thought, and then an additional thought based on the original thought. A chain of thinking rarely jumps from one thought to an unrelated thought. The tagging process is analogous to a chain of thought. When the tags are listed by the order entered, the adjacent tags are more related to each other. Analogous to a bigram approach in natural language processing, in a social tagging system, only adjacent tags are considered to calculate the bigram tag co-occurrence frequency. When a user enters a list of tags (t_1,t_2,t_3,t_4) in turn to annotate an item, a bigram method counts (t_1,t_2), (t_2,t_3), and (t_3,t_4) as bigram pairs. Under the same conditions, a co-occurrence method counts out (t_1,t_2), (t_1,t_3), (t_1,t_4), (t_2,t_3), (t_2,t_4), and (t_3,t_4) as co-occurred pairs. A bigram method considers tag t_1 to be related more to tag t_2 than tag t_4. A co-occurrence method considers all tags in a tag assignment as equally related to each other. If the number of tags in a tag assignment increases, the relevance between the first tag and the last tag will decrease. The co-occurrence method does not consider the decrement of relevance of the tag assignment. For instance, when a user enters a list of tags (*DBMS*, *JDBC*, *Java*, *Programming Language*, *Computer Science*), the adjacent tags *JDBC* and *Java* are more related than other tag pairs that are not formed by adjacent tags.

B_{score} is the relatedness score between a set of user-entered tags, ts, and the candidate tag, t_c, and is defined as follows:

$$B_{score}(ts, t_c) = \frac{1}{N_{ts}} \sum_{t_u \in ts} \frac{P_b(t_u, t_c)}{P(t_c)}$$

N_{ts} is the number of tags in ts, and t_u is the tag in the user-entered tag set, ts. $P_b(t_u,t_c)$ is the probability of a bigram pair, and $P(t_c)$ is the probability of t_c. B_{score} is the average probability of the bigram conditional probabilities between ts and t_c. For instance, when the association rule is *iPhone*, *Steve Jobs* → *Apple*, the t_u tags are *iPhone* and *Steve Jobs*, and t_c is *Apple*. B_{score} of the association rule is then calculated as follows:

$$\frac{1}{2} \times \left(\frac{P_b(iPhone, Apple)}{P(Apple)} + \frac{P_b(Steve\ Jobs, Apple)}{P(Apple)} \right)$$

B_{score} implies that the more tag t_c appears adjacently with the tags in ts, the more t_c is related with the tags in ts.

3.1.3. T_{score} and the recommendation process

T_{score} (r_b,ts,t_c) is the relevance score of the candidate tag t_c given user-entered tag set ts and the best association rule r_b. T_{score} and r_b can be defined as follows:

$$lT_{score}(r_b, ts, t_c) = A_{score}(r_b) \times B_{score}(ts, t_c)$$

$$r_b = \underset{r}{argmax} \left(A_{score}(r) \times B_{score}(ts, t_c) \right)$$

The process of the proposed tag recommendation system is as follows. When a user enters a few tags, the system searches for all association rules that satisfy the sufficient condition. When an association rule satisfies the sufficient condition, the left side of the association rule is a subset of the user-entered tag set ts. For instance, if a user enters two tags, *iPhone* and *Steve Jobs*, there are some association rules that satisfy the sufficient condition, including *iPhone* → *Apple*, *iPhone*, *Steve Jobs* → *Apple*, and *Steve Jobs* → *Apple*. However, the association rule *Apple* → *Pie* is not included because *Apple* is not a subset of the user-entered tags, *iPhone* and *Steve Jobs*.

After searching for appropriate association rules, the next step is selecting the candidate t_c tags for a recommendation. An association rule $A \rightarrow B$ indicates that if tag A exists in the user's tag set, tag B might appear in the user's tag set. Therefore, the candidate t_c tags are on the right side of the searched association rules. When the user enters the tags on the left side of the association rule, the tags on the right side are candidate tags for a recommendation.

For each candidate tag t_c, B_{score} is calculated based on the user-entered tag set ts and candidate tag t_c. To evaluate the final score of the candidate tags, the system searches for the best association rule r_b that maximizes T_{score}. The system then evaluates the T_{score} of all candidate tags and provides a ranked list of candidate tags ordered by T_{score}.

3.2. Item recommendation

Item recommendation is a task in which items highly related to a user are recommended. Correct recommendations are becoming increasingly important owing to an information overload. To alleviate information overload, in this section, we utilize an enriched tagging dataset to recommend items, and describe the proposed item recommendation approach.

As mentioned in Section 2.2, item recommendation can be classified into three categories: a content-based approach, a collaborative filtering, and a hybrid approach. A content-based approach investigates the item's content. Collaborative filtering searches for similar users whose preferences are analogous to the given user's preference. A hybrid approach integrates a content-based approach and a collaborative filtering approach. We previously proposed a content-based method (Kim & Kim, 2012b) and a collaborative filtering method (Kim & Kim, 2012a) for social tagging systems. In this section, we represent a hybrid approach that integrates the proposed content-based method and a collaborative filtering method to mitigate the limitations of each approach. We elucidate each method in Sections 3.2.1, 3.2.2, and 3.2.3, respectively.

3.2.1. Tag expansion

In social tagging systems, a cold-start problem occurs in which a recommender system cannot recommend items relevant to users who have insufficiently tagged items. To alleviate a cold-start problem, we expand the initial tag set of the user. The user's initial tag set is derived from the user's tagging history, and is comprised of all distinct tags utilized by the user.

We utilize a bigram approach to expand the user's tag set. A bigram approach counts out the number of bigram pairs of tags from the folksonomy. By evaluating the number of bigram pairs, tags relevant to the user's initial tag set are expanded to the user's tag set. When tags t_1 and t_2 are paired with a bigram, and a user uses tag t_1, the system adds tag t_2 to the user's tag set. Nonetheless, it is not reasonable to add all tags that are paired with tag t_1. We adopt a conditional probability to solve this problem.

$$\mathrm{P}(t_1|t_2) = \frac{P(t_1, t_2)}{P(t_1)}$$

$P(t_1)$ is the probability of tag t_1, and is the number of t_1 tags in the dataset. $P(t_1,t_2)$ is the probability of a bigram pair (t_1,t_2), and is the number of bigram pairs (t_1,t_2) in the dataset. The expanded tag set of user u, $TS_e(u)$, can be defined as follows:

$$TS_e(u) = \{t_n|\mathrm{P}(t_n|t_l) > \tau, t_l \in TS_i(u)\}$$

$TS_i(u)$ is the initial tag set of user u, t_l is a tag in $TS_i(u)$, and τ is a threshold value. $TS_e(u)$ is the set of tags, t_n, whose conditional probability is greater than threshold value τ. To expand the tag set, tag pair (t_l,t_n) should be a bigram in the folksonomy. If the value of τ is too high, no tags are selected for an expanded tag set. If the value of τ is too low, most tags are selected for the expanded tag set. We empirically determine the optimum value of τ. Expanded tags of user u are added to the initial tag set, and the tag set is utilized as user u's new profile to generate recommendable items.

We propose a method for recommending items to a user with the user's expanded tag set. Given a user's tag set, the system expands the tag set using a bigram method, and the system recommends relevant items related with the user's preference. Each tag in a user's tag set has its own score. A tag score is the probability of a tag utilized by a user. If the user annotates a tag once, the tag score will be low. If the user annotates a tag many times, the tag score will be high. Given user u and tag t, the score of tag t, $w_{u,t}$, can be defined as follows (Zheng & Li, 2011):

$$w_{u,t} = \frac{freq(u,t)}{\sum_{j=1}^{k} freq(u,t_j)}$$

K is the number of distinct tags annotated by user u. $freq(u,t)$ is the frequency of tag t in user u's tagging history. To summarize, $w_{u,t}$ is the proportion of tag t in user u's total tag usage. $STS(u)$ is the scored tag set of user u and can be defined as follows:

$$STS(u) = \{\langle t_m, w_{u,t_m}\rangle | t_m \in (TS_i(\mathrm{u}) \cup TS_e(\mathrm{u}))\}$$

If a tag is frequently annotated by the user, the tag affects the user preference more than infrequently annotated tags. The more a user annotates a tag, the higher the score that is obtained by that tag. Given user u, the system recommends items to user u in order of relevance. The relevance between user u and item p, $R(u,p)$, is defined as the relevance between the scored tag set of user u, $STS(u)$, and the tags in item p.

$$R(u,p) = \sum_{t_l \in MS(p)} \frac{w_{u,t_l}}{N_{nz}}$$

$MS(p)$ is a multiset of tags of item p. An identical tag may appear more than once in the multiset. N_{nz} is the number of tags whose tag score, w_{u,t_l}, is not zero for item p. To summarize, $R(u,p)$ is the average score of tags that exist for both the expanded tag set and item p of user u. In a collaborative tagging model, many users can annotate tags to an identical item. This feature enables the tags to build a folksonomy of the item. If a certain tag is annotated a lot more than other tags, the tag is likely to represent the item precisely. The relevance metric generates a score that reflects the popularity factor of the tag. For instance, when the scored tag set of user u is $\{\langle t_1,0.2\rangle, \langle t_2,0.3\rangle, \langle t_3,0.5\rangle\}$, and three other users annotate the tags to item p, (t_1,t_2), (t_2,t_4,t_5), and (t_1,t_3,t_6), the relevance between user u and item p is evaluated as follows:

$$R(u,p) = \frac{0.2 + 0.3}{5} + \frac{0.3}{5} + \frac{0.2 + 0.5}{5} = 0.3$$

N_{nz} is 5 because $MS(p)$ is $\{t_1,t_2,t_2,t_4,t_5,t_1,t_3,t_6\}$, the expanded tag set of user u is $\{t_1,t_2,t_3\}$, and the scores of t_4, t_5, and t_6 are zero. When two items have an identical tag set over user u, the system cannot discriminate between the two items. For instance, when the expanded tag set of user u is $\{t_1,t_2,t_3\}$, $MS(p_1)$ is $\{t_1,t_2,t_3,t_4,t_5,t_6\}$, and $MS(p_2)$ is $\{t_1,t_2,t_3,t_7,t_8,t_9\}$, the relevance between user u and item p_1, $R(u,p_1)$, is equal to the relevance between user u and item p_2, $R(u,p_2)$. We adopt temporal information to discriminate two items that have an identical tag set over the user. Dai and Davison proved that freshness matters in Web authority (Dai & Davison, 2010). Freshness also matters for item recommendations. If two items have the same score, we rank them in order of freshness. An item that was recently created is ranked higher than an item that was created earlier. The temporal information alleviates the cold-start user and item problems. For cold-start users, recommended items most likely have the same scores because the number of tags of the cold-start user is relatively small. When cold-start items are recommended to users, they are likely to have the same scores because cold-start items have only a few annotations. Under this type of situation, temporal information plays a role as a tiebreaker.

A recommendation method using tag expansion is a type of content-based approach. Such an approach utilizes the tagging information of items as their content and metadata to generate a ranked item list. The process of item recommendation is as follows: (i) add an expanded tag set to the user's initial tag set, (ii) evaluate the scores of the tags in the tag set, and (iii) generate a ranked list of items based on the score metrics.

3.2.2. Trust relationship

To generate recommendable items, a conventional collaborative filtering approach searches for similar users and recommends their items. Collaborative filtering is one of the most successful approaches, and is based on user similarity. User similarity has a symmetric relationship. If user u_a is analogous to user u_b, then user u_b is also analogous to user u_a. Instead of searching for similar users, in this approach, we investigate the trust relationship between users. Trust is "a subject expectation an agent has about another's future behavior based on the history of their encounters" (Mui, Mohtashemi, & Halberstadt, 2002). A trust relationship has an asymmetric relationship. When user u_a trusts user u_b, it does not necessarily imply that user u_b trusts user u_a. User u_b might or might not trust user u_a. An asymmetric relationship enables the propagation of a trust relationship. This feature mitigates data sparsity. A trust relationship depends on the user's personal preference. A group of users may trust a certain user u, while another group of users does not trust user u at the same time.

There are two approaches used to utilize a trust relationship in recommender systems. The first approach utilizes an explicit trust relationship from the user profiles, and the second approach extracts an implicit trust relationship. In the second approach, the trust relationship is extracted from a user profile, rated items, ratings, or tagging information. Most recommender systems do not provide a trust feature in their systems, and we therefore propose a method to elicit implicit trust relationships and perform a recommendation using the trust information.

We utilize triples $\langle user, item, tag \rangle$ of the tagging information. A naïve approach to derive implicit trust information between two users is to calculate the conditional probability of the items or tags. Additionally, an advanced approach used to derive an implicit trust relationship is utilizing a Kullback–Leibler (KL) divergence (Kullback & Leibler, 1951).

A conditional probability is inherently asymmetric, and it is therefore apposite to evaluate an asymmetric relationship between two users. Using the tagged items of users, trust information from user u_a to user u_b, $trust^i_{u_a \to u_b}$, can be defined as

$$trust^i_{u_a \to u_b} = \frac{P_i(u_a \cap u_b)}{P_i(u_a)}$$

where $P_i(u_a)$ is the probability of the items of user u_a, and $P_i(u_a \cap u_b)$ is the probability of items that both u_a and u_b are interested in. Using the user tags, trust information from user u_a to user u_b, $trust^t_{u_a \to u_b}$, can be defined as

$$trust^t_{u_a \to u_b} = \frac{P_t(u_a \cap u_b)}{P_t(u_a)}$$

where $P_t(u_a)$ is the probability of the tags of user u_a, and $P_t(u_a \cap u_b)$ is the probability of tags that are annotated by both u_a and u_b.

KL divergence is more apposite in evaluating asymmetric relationships between two users than conditional probability. The conditional probability only takes the existence of tagged items and tags into consideration. However, KL divergence takes not only the existence of tagged items and tags into consideration, but also their frequency. This consideration enables a detailed analysis of the user preference. KL divergence is a weighted version of a conditional probability.

In a KL-divergence-based approach, using the tagged items of users, trust information from user u_a to user u_b, $D^i_{KL}(u_b||u_a)$, can be defined as

$$D^i_{KL}(u_b||u_a) = \sum_{k \in I(u_a)} f_i(u_b, k) \log \frac{f_i(u_b, k)}{f_i(u_a, k)}$$

where $I(u_a)$ is a set of tagged items that u_a is interested in, and $f_i(u_a, k)$ and $f_i(u_b, k)$ are the probability mass functions of user u_a and user u_b over the tagged items, respectively, and can be defined as follows:

$$f_i(u_a, k) = \frac{n_i(u_a, k)}{\sum_{l \in I(u_a)} n_i(u_a, l)}$$

Here, $n_i(u_a,k)$ and $n_i(u_a,l)$ are the distribution of tagged items k and l of user u_a, respectively. In a KL-divergence-based approach, using the user tags, trust information from user u_a to user u_b, $D^t_{KL}(u_b||u_a)$, can be defined as

$$D^t_{KL}(u_b||u_a) = \sum_{s \in T(u_a)} f_t(u_b, s) \log \frac{f_t(u_b, s)}{f_t(u_a, s)}$$

where $T(u_a)$ is a set of tags annotated by user u_a, and $f_t(u_a, s)$ and $f_t(u_b, s)$ are the probability mass functions of users u_a and u_b over the tags, respectively, and can be defined as follows:

$$f_t(u_a, s) = \frac{n_t(u_a, s)}{\sum_{r \in T(u_a)} n_t(u_a, r)}$$

Here, $n_t(u_a,s)$ and $n_t(u_a,r)$ are the distribution of tags s and r of user u_a, respectively. The system evaluates the trust relationship between two users using KL divergence. Given a user's tagging information, the system elicits an implicit trust relationship from the user's tagging information. The recommendation method using a trust relationship is a type of collaborative filtering approach. This approach searches for candidate users not based on user similarity but based on the trust relationship between users. The process of item recommendation is as follows: (i) evaluate the trust relationship using both the tagged item and tag of the user, (ii) search for other users highly trusted by the given user based on the implicit trust relationships, (iii) aggregate items of the trusted users into a ranked list, and (iv) recommend items in order of their relevance.

3.2.3. Hybrid approach

There are different ways to combine a content-based method and a collaborative filtering method into a hybrid approach (Adomavicius & Tuzhilin, 2005). To achieve a hybridization of the proposed approaches, we combine separate recommendation methods. A method using tag expansion as a content-based method generates a ranked list of items. Another method using implicit trust relationship of items and tags that is a collaborative filtering method also generates a ranked list of items. The hybrid approach aggregates these ranked lists of items. We propose hybrid methods comprising linear combination, voting, and weighted voting. First hybrid approach is linear combination of two methods.

$$score_{lc}(p) = \tau \times s_{TE}(p) + (1 - \tau)$$

The approach adds two weighted scores of the proposed approaches. $s_{TE}(p)$ is the score of tag expansion algorithm and $s_{TR}(p)$ is the score of trust relationship based approach. Hybrid score of item p, $score_{lc}(p)$ is weighted sum of two methods by threshold τ. The optimal threshold τ is selected empirically. We also propose a voting strategy, i.e., a demographic strategy, along with a weighted voting strategy. In voting and weighted voting methods, we utilize a relative ranking of the item in each list. For the voting mechanism, given the user, the system generates three ranked lists of items using the previously mentioned methods. The system then merges each score into the final score of item p, $score_h(p)$, which can be defined as

$$score_h(p) = \sum_{l \in L} s_v(l, p)$$

$$s_v(l, p) = \begin{cases} 1 & \text{if } p \in l \\ 0 & \text{otherwise} \end{cases}$$

where l is a ranked list of items, and L is the set of resulting l lists. The voting score, $s_v(l,p)$, is 1 if item p exists in ranked list l, and *0* if it does not. The more item p appears in the recommendation results, the higher the score of item p. In the weighted voting mechanism, the system merges each score into the final score of item p, $score_{wh}(p)$, which can be defined as

$$score_{wh}(p) = \sum_{l \in L} s_{wv}(l,p)$$

$$s_{wv}(l,p) = \begin{cases} \frac{1}{r} & \text{if } p \in l \\ 0 & \text{otherwise} \end{cases}$$

where r is the rank of item p in ranked list l, and $1/r$ is the reciprocal rank of item p. Therefore, a high value of the reciprocal rank of an item indicates a high relevance of the item with the user. The weighted voting score, $s_{wv}(l,p)$, is $1/r$ if item p exists in ranked list l, and *0* if it does not. The more item p appears in the recommendation results, the higher the score of item *p*. Furthermore, highly ranked items obtain a higher score than lower ranked items. The process of recommendation is as follows: (i) generate ranked lists of items, (ii) aggregate a ranked list through a scoring strategy, (iii) generate a final ranked list of items, and (iv) recommend items in order of their scores.

4. Evaluation

In this section, we describe our experimental evaluation and discuss the results of the proposed approaches. We also analyze the data distribution and the recommendation performance. We conducted a series of experiments using the MovieLens dataset. In the dataset, the users annotate tags to different movies. The dataset provides triples of the users' movie annotations. A total of 7601 movies, 4009 users, 16,529 distinct tags, and 95,580 triples are used.

4.1. Data analysis

From the dataset, we evaluate the user distribution over the number of distinct tags used and the user distribution over the number of distinct movies.

Fig. 2 shows the user distribution over the number of distinct tags. Point (n_t, n_u) on the graph indicates that the number of users who use n_t distinct tags is n_u. This distribution follows the power law. In total, 84.6% of the users used ten or fewer distinct tags, and 39.7% of the users used one distinct tag. In addition, only 1.6% of the users used more than 100 distinct tags.

Fig. 3 shows the user distribution over the number of distinct movies. Point (n_m, n_u) on the graph indicates that the number of users who annotate n_m distinct movies is n_u. This distribution also follows the power law. In this case, 83.8% of the users annotated ten or fewer distinct movies, and 43.3% of the users annotated one distinct movie. In addition, only 2.4% of the users annotated more than 100 distinct movies.

Fig. 4 shows the movie distribution over the number of users. Point (n_u, n_m) on the graph indicates that the number of movies that are annotated by n_u users is n_m. This distribution also follows the power law. Here, 82.1% of the movies are annotated by ten or fewer users, and 24.7% of the movies are annotated by one user. Additionally, only 0.1% of the movies are annotated by more than 100 users. These three data distributions denote that most users are cold-start users and most movies are cold-start items.

We also discovered that a sparsity problem arises in a real dataset. Table 1 shows the statistics of the dataset. In the *user-movie* matrix, the ratio of non-zero terms is 1.820×10^{-3}. In the *user-movie-tag* tensor, the ratio of non-zero terms is 2.043×10^{-7}. The matrix and tensor are very sparse. Even the most active user annotates 21.4% of all movies, and the average value for all users is only 0.18%. The most active user uses 12.5% of all distinct tags, and the average value for all users is only 0.066%. We determined that cold-start and sparsity problems occur in a real-world dataset. Under this situation, a matrix factorization or tensor factorization method will not recommend items properly.

4.2. Performance analysis

We evaluated the recommendation performance of the proposed hybrid approach. We divided a dataset into a training set and a test set. The system generates a ranked list of movies given a user's training set. The recommendation results are evaluated based on the precision and recall values using the user's test set.

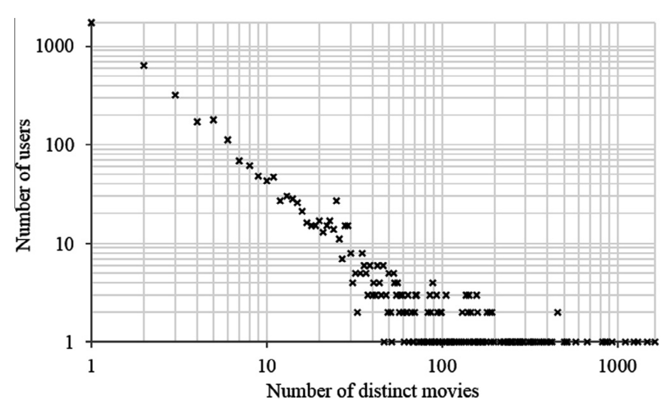

Fig. 3. User distribution over the number of movies.

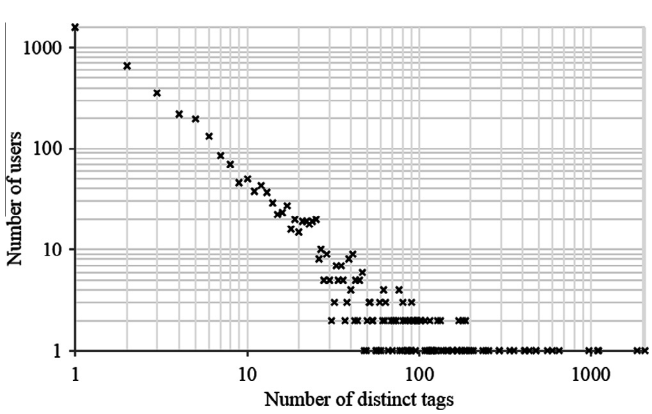

Fig. 2. User distribution over the number of tags.

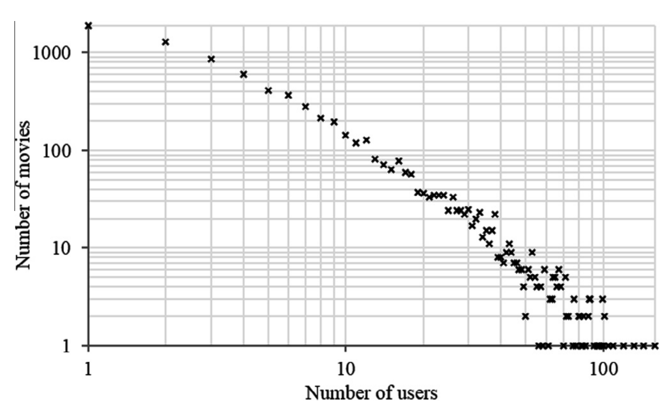

Fig. 4. Movie distribution over the number of users.

Table 1
Sparse matrix and tensor.

	Non-zero term	Active user (%)	Average (%)
(*user-item*) matrix	1.820×10^{-3}	21.4	0.18
(*user-item-tag*) tensor	2.043×10^{-7}	12.5	0.066

First experimental evaluation is conducted to find appropriate threshold τ in linear combination method on the dataset. We randomly choose 100 users from the dataset and each recommendation method generates recommendable items for the users.

Table 2 and Fig. 5 represent the resulting recommendation accuracy of linear combination. In Table 2, recall values are relatively lower than precision values because the number of items of the users is comparatively larger than the number of recommended items. In Fig. 5, x-axis is the threshold value τ and y-axis is the precision value of recommended items. When τ is zero, the system generates recommendable items using trust relationship based approach. When τ is one, the system generates recommendable items using tag expansion approach. When τ is greater than zero and less than one, the system generates recommendable items using both tag expansion and trust relationship based approaches. Evaluation result shows that linear combination of two methods performs better than a single method. Especially, when the threshold τ is 0.4, the system generates the best result on the dataset. The threshold value is not optimum for other datasets and depends on the characteristic and user distribution on the datasets.

Second experiment is conducted to evaluate influence of the number of trusted users on recommendation accuracy. In this evaluation, we focus on cold-start users who annotated ten or fewer movies.

Table 2
Recommendation accuracy of linear combination.

τ	Precision	Recall	F-measure
0	0.254	0.0856	0.1280
0.1	0.265	0.0844	0.1280
0.2	0.290	0.0889	0.1361
0.3	0.295	0.0888	0.1365
0.4	0.299	0.0888	0.1369
0.5	0.294	0.0873	0.1346
0.6	0.292	0.0864	0.1330
0.7	0.291	0.0861	0.1328
0.8	0.291	0.0852	0.1318
0.9	0.290	0.0847	0.1311
1	0.288	0.0831	0.1290

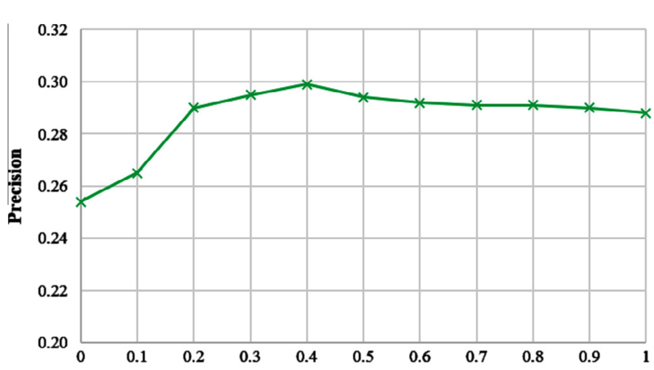

Fig. 5. Linear combination of tag expansion and trust relationship based approaches.

Table 3
Recommendation accuracy over the number of trusted users.

	n	CB	$TR(t)$	$TR(i)$	$TR(t+i)$	$CB+TR(t+i)$
Precision	5	0.130	0.038	0.181	0.188	0.211
	10	0.130	0.046	0.203	0.208	0.226
	20	0.130	0.047	0.204	0.207	0.225
	30	0.130	0.047	0.199	0.202	0.220
	40	0.130	0.046	0.195	0.198	0.217
	50	0.130	0.045	0.190	0.193	0.212
Recall	5	0.253	0.100	0.540	0.558	0.618
	10	0.253	0.121	0.599	0.611	0.659
	20	0.253	0.127	0.597	0.606	0.651
	30	0.253	0.128	0.584	0.591	0.640
	40	0.253	0.123	0.575	0.582	0.633
	50	0.253	0.103	0.472	0.569	0.620
F-measure	5	0.172	0.055	0.271	0.281	0.315
	10	0.172	0.067	0.303	0.310	0.337
	20	0.172	0.069	0.304	0.309	0.334
	30	0.172	0.069	0.297	0.301	0.327
	40	0.172	0.067	0.291	0.295	0.323
	50	0.172	0.063	0.271	0.288	0.316

Table 3 and Fig. 6 represent the resulting recommendation accuracy on various number of trusted users. In Table 3, recall values are relatively higher than precision values because the number of items of the user is small. In Fig. 6, *CB* is an abbreviation for a content-based approach, and indicates the tag expansion approach described in Section 3.2.1. *TR* is an abbreviation for a trust relationship approach, and indicates the trust-relationship-based approach described in Section 3.2.2. The $TR(t)$ approach searches for trusted users using user tags. The $TR(i)$ approach searches for trusted users using the user's tagged items. The $TR(t+i)$ approach is a hybrid approach of $TR(t)$ and $TR(i)$. The $CB+TR(t+i)$ approach is a hybrid approach of *CB* and $TR(t+i)$. In the hybrid approaches, each weak recommender generates a ranked list of items, and the lists are merged into a final ranked list of recommendations. The x-axis in Fig. 6 shows the number of trusted users. When the *TR* approach generates recommendable items, a process for searching n trusted users is necessary. We conducted experiments on various values of n: 5, 10, 20, 30, 40, and 50. The y-axis in Fig. 6 shows the value of the F-measure. Fig. 6 shows that the smallest value of the F-measure is represented by $TR(t)$. This method considers user tags to search for trusted users. This method, which does not utilize an item's content or a user's tagged item, cannot achieve a high recommendation accuracy because the system recommends items not tags. The *CB* approach results in uniform F-measure values because the method is not affected by the number of trusted users. The hybrid approaches, $TR(t+i)$ and $CB+TR(t+i)$, show the best results. The results show that, when a content-based approach and collaborative filtering approach are hybridized, the limitations of both approaches are alleviated. If the number of trusted users is from ten to twenty, the recommendation accuracy is the highest. As the number of trusted users increases up to fifty, the performance decreases slightly. The result indicates that searching for many trusted or similar users does not necessarily result in a better performance.

In the next experimental evaluation, we compare the performance of the proposed hybrid approach (H) with the following methods. First, popularity-based method (P) recommends items with popularity of the item in the dataset. The most tagged item is recommended first. Second, user-based collaborative filtering (U) is one of the most successful types of recommendation method. We choose cosine similarity for calculating affinity between users. Third, tensor factorization method (T) (Karatzoglou et al., 2010) decomposes user-item-tag tensor into three matrices and one core tensor. The proposed n-dimensional tensor factorization model is for context-aware collaborative filtering. We assume that the tagging information is an additional context in the model. The model

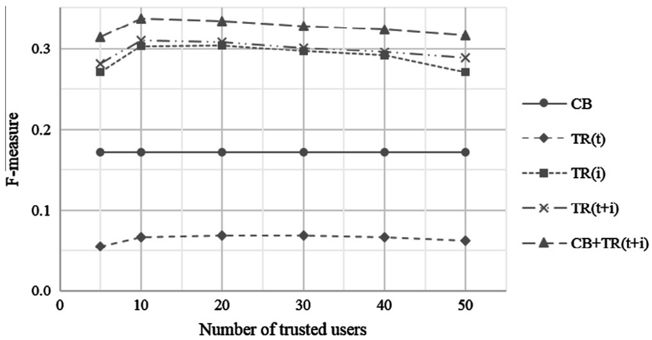

Fig. 6. Recommendation performance of the proposed hybrid approach.

conjectures latent values of the tensor. We choose squared error as the loss function and choose stochastic gradient descent (SGD) to minimize objective function. We set the regularization parameter $\lambda = 0.001$ and dimensions of core tensor $d_U = d_M = d_C$.

We conduct experimental evaluation by leave-one-out and re-ranking method (Cremonesi, Koren, & Turrin, 2010). For given user, there are the answer set of the user and randomly selected 1000 items. Randomly selected items are not tagged by the user. Each recommendation method generates a ranked list of these items in company with items in the answer set. Then, based on the ranking of the items in the answer set, the precision and recall values are measured for performance evaluation. In the testing methodology, precision is proportional to recall.

To capture the effectiveness of the proposed approach in detail, we group users by activeness and conduct experiments for each group. All users in the dataset are in user group 1 (UG_1). Users who annotated less than or equal to ten items are in UG_2 and users who annotated greater than ten items are in UG_3. Users who used less than or equal to ten distinct tags are in UG_4 and users who used greater than ten distinct tags are in UG_5. Fig. 7 shows the recall values of each methods for each user group. For all users (UG_1), the performance of the proposed approach (H) exceeds tensor factorization (T) by 1.6%. For UG_2 and UG_4, i.e., cold-start users, the gap between the proposed approach and other method increases. The proposed approach performs best in UG_4 because the proposed approach utilizes tag expansion and trust relationship to alleviate cold-start and sparsity. When a user does not provide enough information to form the user's profile, the result shows that the sparseness and binary values of a tensor decreases the accuracy of learning in one single model. As we mentioned in Section 4.1, the ratio of non-zero term of *user-item-tag* tensor is very small. Using the information, tensor factorization method conducts factorization and conjectures a large amount of unseen latent values in the tensor. Moreover, in social tagging system, the tensor consists of binary values. If a user annotates an item with a tag, then the value is one, otherwise zero. The simple binary values may decrease the accuracy of tensor factorization. For UG_3 and UG_5, i.e., active users, the proposed approach performs worse than tensor factorization method. The result shows that tag expansion decreases the accuracy of recommendation because active users already have enough tags. Therefore, the expanded tags are noisy tags for the users.

In sum, the proposed hybrid approach alleviates cold-start and sparsity problems without losing overall accuracy.

5. Conclusion

In this paper, we propose a hybrid item recommendation and a recommendation framework for social tagging systems. In the hybrid method, we utilize tag expansion and implicit trust relationship at the same time. Tag expansion method is more useful to cold-start users and implicit trust relationship from tagging information reflects asymmetric relationship of user similarity. The hybrid method mitigates limitations of the two approaches. Tag expansion alleviates cold-start problem and the recommendation result based on implicit trust relationship reduces over-specification. On real-world social tagging dataset, we conduct experimental evaluations of the proposed methods to search for optimum threshold value, to verify influence of the number of trusted users on accuracy, and to compare the proposed approach against existing algorithms. For less active users, as we expected, the hybrid approach performs better than other methods. In the proposed framework, we contemplated tag and item recommendations in social tagging systems. Tag recommendation generates high quality tagging information then item recommendation applies stored tags to generated recommendable items. Moreover, the proposed approach is applicable to social tagging systems because recommendation methods in the framework do not require content of items. By analyzing tagging information on the system, it can be applied to any kinds of social tagging systems that share textual and non-textual items including bookmarks, photos, videos, news articles, and research papers.

As future work, we will tackle the scalability problem in recommender systems. The amount of data in social tagging systems is increasing significantly. For example, a crawled social tagging dataset includes more than 1 billion triples and 4.5 billion tag co-occurrence pairs. Given situation, computations of association rule mining, bigram tag co-occurrence frequency, conditional probability, and KL divergence are time-consuming tasks. As the

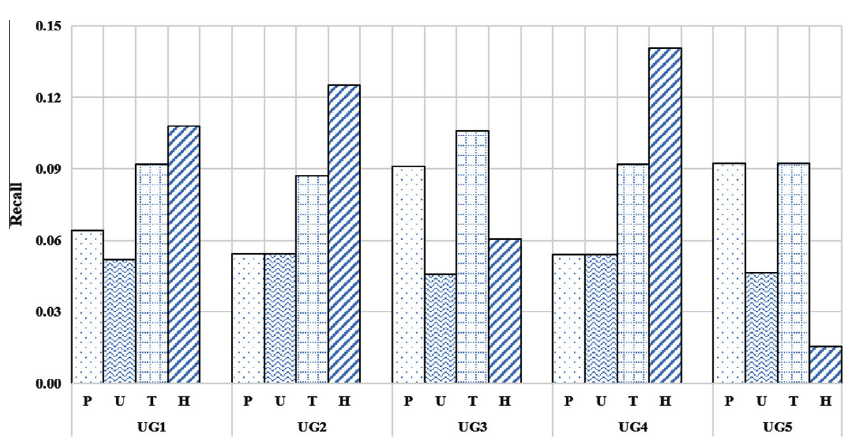

Fig. 7. Comparison of recommendation methods.

size of data increases, not only generating data for recommender systems but also the process of recommendation on a single node will not terminate on feasible execution time. For scalable recommendation in social tagging systems, we will propose a parallel computation method on distributed environment such as MapReduce framework. We also plan to improve recommendation accuracy for active users. In evaluation, active user's expanded tags decreases performance. By analyzing user profiles, dynamic application of hybrid method will mitigate the problem.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 20120005695).

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*, 734–749.
Ames, M., & Naaman, M. (2007). Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 971–980). ACM.
Cai, Y., Zhang, M., Luo, D., Ding, C., & Chakravarthy, S. (2011). Low-order tensor decompositions for social tagging recommendation. In *Proceedings of the fourth ACM international conference on web search and data mining* (pp. 695–704). ACM.
Chirita, P.-A., Costache, S., Nejdl, W., & Handschuh, S. (2007). P-tag: large scale automatic generation of personalized annotation tags for the web. In *Proceedings of the 16th international conference on world wide web* (pp. 845–854). ACM.
Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on recommender systems* (pp. 39–46). ACM.
Dai, N., & Davison, B. D. (2010). Freshness matters: in flowers, food, and web authority. In *Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval* (pp. 114–121). ACM.
Das, M., Thirumuruganathan, S., Amer-Yahia, S., Das, G., & Yu, C. (2012). Who tags what? an analysis framework. *Proceedings of the VLDB Endowment, 5*, 1567–1578.
Garg, N., & Weber, I. (2008). Personalized, interactive tag recommendation for flickr. In *Proceedings of the 2008 ACM conference on recommender systems* (pp. 67–74). ACM.
Golder, S. A., & Huberman, B. A. (2006). Usage patterns of collaborative tagging systems. *Journal of Information Science, 32*, 198–208.
Guan, Z., Wang, C., Bu, J., Chen, C., Yang, K., Cai, D., et al. (2010). Document recommendation in social tagging services. In *Proceedings of the 19th international conference on world wide web* (pp. 391–400). ACM.
Guy, I., Zwerdling, N., Ronen, I., Carmel, D., & Uziel, E. (2010). Social media recommendation based on people and tags. In *Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval* (pp. 194–201). ACM.
Han, H., Cai, Y., Shao, Y., & Li, Q. (2012). Improving recommendation based on features' co-occurrence effects in collaborative tagging systems. In *Proceedings of the 4th asia-pacific web conference* (pp. 652–659). Springer.
Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*, 5–53.
Karatzoglou, A., Amatriain, X., Baltrunas, L., & Oliver, N. (2010). Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on recommender systems* (pp. 79–86). ACM.
Kim, H., & Kim, H. -J. (2012a). Improving recommendation based on implicit trust relationships from tags. In *Proceedings of the 2nd international conference on computers, networks, systems, and industrial applications* (pp. 25–30).
Kim, H., & Kim, H.-J. (2012b). Item recommendation using tag expansion and temporal information. *Journal of KIISE: Computing Practices and Letters, 18*, 521–527.
Kim, H., Lee, K., Shin, H., & Kim, H.-J. (2009). Tag suggestion method based on association pattern and bigram approach. In *Proceedings of the 10th ACIS international conference on software engineering, artificial intelligences, networking and parallel/distributed computing* (pp. 63–68). IEEE.
Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics, 22*, 79–86.
Lee, K., Kim, H., Shin, H., & Kim, H.-J. (2009). Tag sense disambiguation for clarifying the vocabulary of social tags. In *International conference on computational science and engineering* (4, pp. 729–734). IEEE.
Lee, S., Yang, J., & Park, S.-Y. (2004). *Discovery of hidden similarity on collaborative filtering to overcome sparsity problem. Discovery Science*. Springer, pp. 396–402.
Lipczak, M. (2008). Tag recommendation for folksonomies oriented towards individual users. In *ECML PKDD discovery challenge Vol. 84*, (pp. 84–95).
Milicevic, A. K., Nanopoulos, A., & Ivanovic, M. (2010). Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review, 33*, 187–209.
Mui, L., Mohtashemi, M., & Halberstadt, A. (2002). A computational model of trust and reputatio. *Proceedings of the 35th annual hawaii international conference on system sciences* (pp. 2431–2439). IEEE.
Peng, J., Zeng, D. D., Zhao, H., & Wang, F.-Y. (2010). Collaborative filtering in social tagging systems based on joint item-tag recommendations. In *Proceedings of the 19th ACM international conference on information and knowledge management* (pp. 809–818). ACM.
Rendle, S., Balby Marinho, L., Nanopoulos, A., & Schmidt-Thieme, L. (2009). Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 727–736). ACM.
Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on computer supported cooperative work* (pp. 175–186). ACM.
Shepitsen, A., Gemmell, J., Mobasher, B., & Burke, R. (2008). Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on recommender systems* (pp. 259–266). ACM.
Sigurbjörnsson, B., & Van Zwol, R. (2008). Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on world wide web* (pp. 327–336). ACM.
Song, Y., Zhang, L., & Giles, C. L. (2011). Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web, 5*, 4.
Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W.-C., et al. (2008). Real-time automatic tag recommendation. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 515–522). ACM.
Symeonidis, P., Nanopoulos, A., & Manolopoulos, Y. (2008). Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM conference on recommender systems* (pp. 43–50). ACM.
Symeonidis, P., Nanopoulos, A., & Manolopoulos, Y. (2010). A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. *IEEE Transactions on Knowledge and Data Engineering, 22*, 179–192.
Wu, L., Yang, L., Yu, N., & Hua, X.-S. (2009). Learning to tag. In *Proceedings of the 18th international conference on world wide web* (pp. 361–370). ACM.
Xu, Z., Fu, Y., Mao, J., & Su, D. (2006). Towards the semantic web: collaborative tag suggestions. In Collaborative web tagging workshop scotland.
Zhao, S., Du, N., Nauerz, A., Zhang, X., Yuan, Q., & Fu, R. (2008). Improved recommendation based on collaborative tagging behaviors. In *Proceedings of the 13th international conference on intelligent user interfaces* (pp. 413–416). ACM.
Zheng, N., & Li, Q. (2011). A recommender system based on tag and time information for social tagging systems. *Expert Systems with Applications, 38*, 4575–4587.