Send Orders for Reprints to reprints@benthamscience.ae

Diversified Coverage based Tag Recommendation

Yuhai Zhao[1], Ying Yin[1,*], Qingze Wang[1] and Gang Sheng[2]

[1]*College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China*

[2]*Software Center, Northeastern University, Shenyang, Liaoning 110004, China*

Abstract: Tag recommendation, as a branch of recommendation engine, has drawn more and more attention, which is also extensively exploited in e-commerce and SNS (Social Networking Services). The results generated by the current algorithms could describe the items with a high relevance. However, they are often of poor diversity in the recommended results. That indicates there is a redundancy in the results in term of semantics. Such a case reduces the novelty and diversity of the recommended results, seriously affecting the user's experience. In this paper, we define the tag correlation metric based on the local and global tag co-occurrence matrices, which improves the recommendation accuracy by incorporating both the user's interests and the popularity of tags. Moreover, we propose the concept of semantic coverage, by which the redundancy of semantics can be removed efficiently. To our best knowledge, it is first proposed in the context of tag recommendation. Finally, a diversified coverage based tag recommendation algorithm, namely EDC, is developed. By converting the problem of diversified coverage tag recommendation to the MIDS (Minimum Independent Dominating Set) problem, EDC first handles the cliques and the bipartites in the graph. Then, it recursively searches the MIDSs in the remaining graph. Further, a greedy algorithm GDC is proposed. The experiments conducted on the real datasets of MovieLens and Last.fm show that the proposed EDC and GDC improve the diversity significantly.

Keywords: Tag recommendation, diversity, coverage, correlation.

1. INTRODUCTION

Network information is increasing at an unimaginable speed every day. The popularity of Web2.0 has accelerated this growth trend to great extent. The massive information published by Web2.0 users not only enriches information content of the Internet, but also speeds up the diffusion of information content on the Internet. This results in information overload of the Internet, an increase in search load, reduction in the quality of information and many other problems.

One of the solutions to information overload is the information retrieval system represented by the search engines [1], such as Google, Baidu, etc. When users search for keywords, the system will return results related to it, filtering irrelevant information. But the problem is that when the same keyword is used for search, the same result will be returned [2]. However, since what the users concern is often not the exactly same, the same result returned is not suitable for them. The infromation demands from different users are often personalized and diverse. Thus, the way that return the same result for different users by search engines cannot meet the personalized needs of users, and cannot provide a good solution to the problem of information overload.

Recommender system is one of the promising solutions towards effectively solving the personalized problem [3]. It understands the practical requirements of the users by collecting the users' interests, and then recommends the information, the products and others, which the users' may be highly interested in. Different from the search engines, the recommender system focuses on studying the interests and the preference of every user. In this way, it conducts the personalized calculation for every different user so as to find the specific interests corresponding to different users. Recommender system enables the personalized content for the individuals and is of an interdependent relationship with the users.

In the personalized recommender engine, tag recommender plays an important role [4]. As a kind of information organization and classification manner, tag is a typical folksonomy. The traditional expert classification is a top-down method, which is a rigidly controlled hierarchy classification. The folksonomy is a completely different way, which is of the following advantages: (1) the tags are produced by the Web 2.0 users when they mark information on the Internet. With spread and shared, these tags ultimately forms a new taxonomy, as is a bottom-up process; (2) different from the traditional expert classification, tag is more flexible. The users can utilize any word or phrase to mark the information content with high flexibility and usability. Moreover, in this way, a recommended item can simultaneously belong to several categories; (3) in the folksonomy, being of the semantics problem, Web 2.0 users can describe a specific information

1874-4443/14

Table 1. An example of q_u.

	t_1	t_2	t_3	t_4
q_u	1	0	0	1

Table 2. An example of L_u.

	m_1	m_2	m_3	m_4
t_1	1	0	1	1
t_2	0	1	1	0
t_3	1	0	1	1
t_4	1	1	1	1
t_5	1	1	0	0
t_6	1	0	0	0
t_7	0	1	0	0

content from several different aspects, as results in the multi-dimensional and multi-level taxonomic structure.

Although the tag-based taxonomy is of the above advantages, it still has some drawbacks: (1) most of the Web 2.0 sites allow users to enter their own preferable tags. However, the arbitrary nature of users' tags may introduce a massive number of noises, such as misspellings, ambiguity and some nonsense words or phrases, which affects the quality of the tags; (2) there is a problem of sparse data in the tag application. As a new information organization and classification way, tags have not been efficiently applied in many sites. This leads to a large number of unknown content in the tag data such that the data is very sparse.

In view of the above issues, two diversified coverage based tag recommendation algorithms are proposed in this paper. The main contributions are as follows: (1) we design a novel correlation measure based on the local and the global tag co-occurrence matrices. By integrating the users' personal interests and the degree of tag recognition, the accuracy of recommendation is improved; (2) we first introduce the semantic coverage into tag recommendation. By using the WordNet dictionary, we define the semantic diversity from the perspective of IC such that the semantic redundancy in the results was effectively alleviated; (3) by mapping the problem of the diversity based tag coverage to the minimum independent dominating set problem, two algorithms, namely EDC and GDC, are developed, respectively; (4) extensive experiments are conducted on two real datasets to verify the efficiency and the effectiveness of the proposed algorithms. The experiment results show that our methods significantly improve the result diversity.

The rest of this paper is organized as follows: Section 2 gives some basic concepts and the problem description; Section 3 introduces a prefilter method to to select a candidate tag set; Section 4 details the two proposed algorithms; The experimental analysis is given in Section 5; Finally, Section 6 concludes this paper.

2. THE PRELIMINARY

2.1. Basic Definitions

This paper studies the diversified coverage based tag recommendation problem. Specially, besides the diversity and the coverage, we also take into account the tag correlation. This is because, without considering the tag correlation, the recommendation results may be of less practical significance or not associated with the items. Therefore, this section first gives the concept of correlation and the corresponding measure, and then introduces the diversified coverage.

Definition 1: Correlation. The semantic correlation degree between an item attribute and a tag is simply sated as correlation. Let $S=\{v_1,v_2,\ldots\}$ be a recommendation result set. Then, $\forall v_i \in S$, $score(v_i)$ denotes the correlation between v_i and an item.

In this paper, we measure the tag correlation by the tag co-occurrence matrix, which describes how the tags occur together. Next, given the number of tags n and the number of items m, we introduce the tag co-occurrence matrix.

Let q_u be a $1\times n$ tag vector of user u, where if user u has used tag t, the corresponding element is set to 1; otherwise, 0. Table **1** is an example of q_u. As seen, the elements corresponding to t_1 and t_4 are 1, and that corresponding to t_2 and t_3 are 0. Thus, we know that the user has used t_1 and t_4, but has not used t_2 and t_3.

Similarly, let L_u be an $n\times m_u$ history record of all the tags having been labeled by a user u, where m_u is the number of items having been labeled by u. L_u is actually an binary matrix, where the values can only be 0 or 1. If an item m_j has been labeled by a tag t_i, the correspondint entry $[i, j]$ is set to 1; otherwise, 0. u is a user demanding the recommendation result. Table **2** is an example of L_u, where each row denotes a tag, and each column denotes an item. As seen from Table **2**, we know that user u has labeled items m_1, m_3 and m_4 using t_1, but has not labeled m_2 using any tag.

Table 3. An example of C_u.

	t_1	t_2	t_3	t_4	t_5	t_6
t_1	0	10	5	1	15	8
t_2	10	0	1	0	1	9
t_3	5	1	0	1	0	7
t_4	1	0	1	0	0	0
t_5	15	1	0	0	0	1
t_6	8	9	7	0	1	0

G is an $n \times m$ matrix like $\boldsymbol{L_u}$, recording the history information of all tags having been used. That is, if an item m_j has been labeled by a tag t_i, the entry $[i,j]$ in G is set to 1; otherwise, 0. However, a slightly different from $\boldsymbol{L_u}$, it is not for a specific user. Instead, it records the labeling information between a tag and an item for any user.

C_u is a tag co-occurrence matrix of all used tags, which is a $m_u \times m_u$ symmetric matrix. Any element $[i,j]$ in C_u denotes the number of times that tags t_i and t_j are together used by user u. C_u can be computed as follows:

$$C_u = L_u L_u^T \tag{1}$$

Table **3** is a tag co-occurrence matrix of a user u, where rows and columns are both tags. As seen from Table **3**, tags t_1 and t_2 occur 10 times together. *Note*: since the co-occurrence for a single tag is meaningless, the diagonal elements of the matrix are all 0s.

C is a tag co-occurrence matrix like C_u. However, C records the co-occurrence of all used tags, the size of which is $n \times n$, but C_u just records the co-occurrence of a specific user, whose size is relatively smaller. Particularly, C can be derived according to Eq. (2). The element $[i,j]$ denotes the number of times that tags t_i and t_j occur together in all users' history usage.

$$C = GG^T \tag{2}$$

Besides the tag co-occurrence matrices mentioned above, another important concept directly related with the tag correlation measurement is TF-ITF, which is inspired by TF-IDF in information retrieval. In TF-IDF, TF is the term frequency, i.e. the number of times that a term t occurs in a document d, which indicates the importance of t to d. IDF is the inverse document frequency, which is the reciprocal of the frequency of t in d. IDF indicates the discrimative power that t distinguishes d from the other documents.

Inspired by TF-IDF, if a tag seldomly occurs with other tags, it should be well utilized to organize and classify the item when it is used to label an item. This is intuitively because that the information contained by such a tag is so particular that it can distinguish the item by itself. In this paper, we introduce TF-ITF in the context of tag recommendation, which can be calculated by Eq. (3).

$$TF \times ITF = \frac{|t|}{|t_{all}|} \times \log\left(\frac{|T|}{|\bigcup_{t \in T_i} T_i| + 1} + 1\right) \tag{3}$$

where $|t|$ is the number of times that t occurs, $|t_{all}|$ is the total number of times that all tags occur, $|T|$ is all the number of distinct tags in the system and $|\bigcup_{t \in T_i} T_i|$ is the number of distinct tags by which the items containing tag t are labeled.

Given the co-occurrence matrix and TF-ITF as above, we give the tag correlation measurement. Particularly, the tag correlation consists of two parts, i.e. the local correlation and the global correlation, where the former indicates the user's preference and the latter indicates the degree of tag prevalence and recognition. The specific measurement is shown in Eq. (4).

$$score(c) = u \sum_{t \in q} wl(t) sl(t,c) + (1-u) \sum_{t \in q} wg(t) sg(t,c) \tag{4}$$

where $wl(t)$ and $wg(t)$ are the local and the global weight of tag t, respectively; $sl(t,c)$ and $sg(t,c)$ are the local and the global correlation of tag t, respectively; q is the set of tags, which have been used by the users; u is a parameter, regulating the importance of the local and the global correlations in Eq. (4).

Definition 2: Similarity. Let v_i and v_j be two recommendation results. For a given threshold τ, we say that v_i and v_j are similar, denoted as $v_i \approx v_j$, if and only if $sim(v_i,v_j) > \tau$. Specially, the similarity function $sim(v_i,v_j)$ is defined by Eq. (5), where $s(v_i)$ (resp. $s(v_j)$) is the set of concepts v_i (resp. v_j) belongs to, and *lso* is the lowest common ancestor of v_i and v_j in WordNet.

$$sim(v_i, v_j) = \frac{1}{|s(v_i)| \times |s(v_j)|} \times \sum_{\forall c_1 \in s(v_i), \forall c_2 \in s(v_j)} max\left(\frac{WIC(lso(c_1,c_2))}{WIC(c_1)}, \frac{WIC(lso(c_1,c_2))}{WIC(c_2)}\right) \tag{5}$$

Note: in a tag recommendation system, most of the tags are arbitrarily marked by the users. This may cause many tags unable to be found in the WordNet. We introduce the method in [5] to address this issue. Let $n(m,t)$ be the times

that item m was tagged by t, $n(t)=\sum_{m} n(m,t)$ is the total number of tag t occurring in the system, and $N(m)=\sum_{t} n(m,t)$ is the number of tags that item m was tagged. The similarity of tags v_i and v_j is defined as follows:

$$sim(v_i,v_j)=\sum_{\forall m\in I} q(v_i\,|\,m)Q(m\,|\,v_j) \tag{6}$$

where I is the set of all items in the system,

$$Q(m\,|\,v_j)=\frac{n(m,v_j)}{n(v_j)} \tag{7}$$

and $$q(v_i\,|\,m)=\frac{n(m,v_i)}{N(m)} \tag{8}$$

Definition 3: Diversity. Let $S=\{v_1,v_2,\ldots\}$ be a recommendation result set. Diversity refers to the averaged pairwise difference of the tags in the recommendation results. The diversity degree of S, denoted as $diver(S)$, is calculated as follows:

$$diver(S)=1-\frac{\sum_{i,j\in S,i\neq j} sim(v_i,v_j)}{\frac{1}{2}|S|(|S|-1)} \tag{9}$$

Definition 4: Semantic Similarity Graph. Let $S=\{v_1,v_2,\ldots\}$ be a recommendation result set. Semantic similarity graph is an undirected graph, denoted as $G(S)=\{V,E\}$, where $\forall v_i\in S$, there is a corresponding node $v_i\in V$ in the G (S). If $v_i\approx v_j$, there is an edge $(v_i,v_j)\in E$. In the case of no confusion, we use v_i to denote $v_i\in S$ and $v_i\in V$, G to denote $G(S)$.

Definition 5: Minimum Independent Dominating Set. Suppose that $G(S)=\{V,E\}$, $V^*\subseteq V$, and $(v_i,v_j)\notin E(v_i,v_j\in V^*)$. If $\forall v_i\notin V^*$, $\exists(v_i,v_j)\in E$, $v_j\in V^*$, then we say that v_i covers v_j, and V^* is the coverage of G, denotes as $V^*=cover(G)$. Specially, in this case, we say that all the elements in V^* are independent with each other. Let $|V^*|=|\{i|i\in V^*\}|$, V^* is referred to as the minimum coverage iff. $|V^*|$ is the minimum, also called a minimum independent dominating set.

2.2. The Problem Statement

In this paper, we address the diversified coverage based tag recommendation problem. The goals include: (1) improving the existing algorithms such that the tag recommendation result is of a better semantic diversity; (2) finding Top-K results, which can cover all the candidate tags in terms of semantics. We address the two problems by converting it into the problem of finding the minimum independent dominating set in the semantic similarity graph.

The problem statement: Given an object (or say an item) o and the corresponding tag set I_o of o, the task of the diversified coverage based tag recommendation is to get a Top-K result set R from the candidate set $S=\{v_1, v_2,\ldots\}$ s.t. the following conditions:

(1) $|R|=K$;

(2) $I_o\cap R=\varnothing$;

(3) $score(R)=\sum_{v_i\in R} score(v_i)$ is maximum;

(4) $diver(R)$ is maximum;

(5) R is the minimum coverage set of S.

3. PREFILTER

To address the problem mentioned in Section 2.2, we first perform a prefilter to select a candidate set of tags, based on which the diversified coverage based searching is conducted. The prefilter consists of two major steps, i.e. correlation computing and candidate set selection.

3.1. Correlation Computing

Diversified coverage is based on correlation. That is, diversified coverage based tags must be correlated with items. Although recommending uncorrelated or lowly correlated tags can improve the result diversity, it is of no significance to the users.

Concretely, the correlation computing process is as follows: First, enter the correlation data of the users, and construct the user tag vector $\boldsymbol{q_u}$, the matrix $\boldsymbol{L_u}$ of the users' tagging history, the tagging matrix $\boldsymbol{G}$ of the tags in the system, the tag co-occurrence matrix $\boldsymbol{C_u}$ of the users and the tag co-occurrence matrix $\boldsymbol{C}$ of the tags in the system. Limited by space, we omit this step in this paper. Instead, directly call Eq. (4) to calculate the correlation, as shown in Algorithm **1**.

Algorithm 1: Cal_relevance()

Input: $\boldsymbol{q_u}$, $\boldsymbol{L_u}$, $\boldsymbol{G}$, $\boldsymbol{C_u}$, $\boldsymbol{C}$, Tag set T
Output: S, the set of candidate tags
1: *for* each t_i in T *do*
2: s←score(t_i); //calling Eq. (4)
3: $S=S\cup\{<t_i, s>\}$;
4: return S

In Algorithm **1**, $\boldsymbol{q_u}$, $\boldsymbol{L_u}$, $\boldsymbol{G}$, $\boldsymbol{C_u}$, $\boldsymbol{C}$ and tag set $\boldsymbol{T}$ are the inputs, where $\boldsymbol{q_u}$ is the user tag vector, $\boldsymbol{L_u}$ is the tag matrix of the users' historical tagging information, $\boldsymbol{G}$ is the tagging matrix of the tags in the system, $\boldsymbol{C_u}$ and $\boldsymbol{C}$ are the tag co-occurrence matrix of the users and the tag co-occurrence matrix of the tags in the system, respectively. For each tag t_i in the tag set T, The EDC algorithm computes the correlation by Eq. (4), and adds it into the set S in the form of $<t_i, s>$, where t_i is a tag, s is the correlation between the tag and the object.

3.2. Candidate Set Selection

Once the tag correlation is computed, some tags can be selected as the candidates by a filtering-and-verification method. In this paper, we do this by an incremental Top-K

framework. In this framework, the recommendation results is generated one by one based on the correlation in a non-increasing order. That is, select the result of the largest correlation every time, and add it to the Top-K result set, until all the K results are obtained. The process are shown in Algorithm **2**. In Algorithm **2**, we first sort all the tags according to the correlation in a non-increasing order. Then, in the *for* statement ranging from 1 to k, if *next*() can return the next correlation result, add the result to R; Otherwise, the algorithm terminates after returning R.

Algorithm 2: Incremental()

Input: k, the size of the result set
Output: R, the result set
1: sort(); //sorting w.r.t the correlation
2: *for* i=1 to k do
3: $v \leftarrow next()$;
4: *if* $v=\varnothing$ *then*
5: break;
6: $R \leftarrow R \cup \{v\}$;
7: return R

4. THE DIVERSIFIED COVERAGE BASED SEARCH

After computing the tag correlation and selecting the candidate result set, we introduce how to perform the diversified coverage based search. Practically, this step consists of two major sub-steps: First, construct a semantic similarity graph by computing the pairwise diversity for the tags in the candidate set; Second, convert the diversified coverage problem to the problem of searching the minimum independent dominating set in the semantic similarity graph. Below, we details these two sub-steps.

4.1. Construction of Semantic Similarity Graph

Due to the filtering-and-verification process, we know that, if the current iteration is not the first one, the semantic similarity graph constructed in the last iteration must already exist in the current iteration. At this time, if there are still some tags added to the candidate set, it just needs to compute the pairwise similarity between the newly added tags or between the new tag and the tag in the candidate result set. Algorithm **3** gives the pseudo code of this process.

In Algorithm **3**, first of all, a two-dimensional symmetric matrix is constructed to store the semantic similarity graph. In Line 2, we check whether there is already a semantic similarity graph. If it exists, the existing matrix is used to fill in the new matrix; otherwise, construct the new matrix from scratch. Next, travers all the tags in the candidate set to calculate the semantic similarity and complete the remaining semantic similarity graph. In Line 6, we check whether the two current tags are both in WordNet. If yes, call *wordnet_sim*(t_i,t_j) in Line 7, where WordNet is utilized to calculate the semantic similarity; otherwise, call s←*co-occurrence_sim*(t_i,t_j) in Line 9, which compute the semantic similarity based on the tag co-occurrence. Line 10 judges if the semantic similarity between the two tags is larger than the similarity threshold τ. If it is larger than τ, we consider that the two tags are similar and set the value of the corresponding position in the semantic similarity matrix G as 1; otherwise, the value is set as 0. That is, the tags are the diversified tags.

Algorithm 3: build-semantic-similarity-graph()

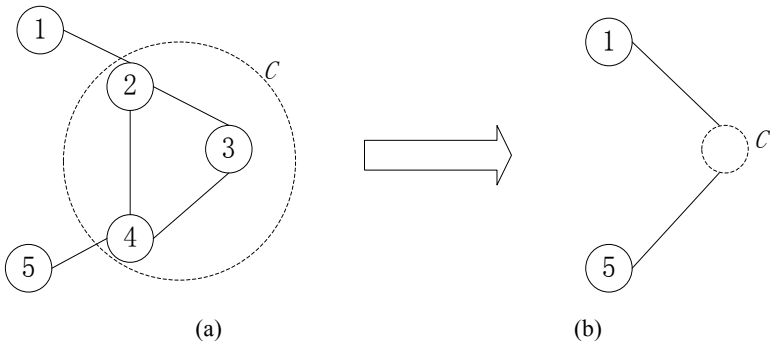
Input: τ, the diversity threshold; G', the existing semantic similarity graph
Output: G, the final semantic similarity graph
1: G = new int[][]; // two-dimensional matrix G
2: $G \leftarrow G'$; // the existing matrix is used to fill in the new matrix
3: for each t_i in S_p do
4: for each $t_j (j<i)$ in S_p do
5: if $G[i][j]$ is not set then
6: if t_i and t_j both in WordNet then
7: $s \leftarrow wordnet\_sim(t_i,t_j)$;
8: else
9: $s \leftarrow co\text{-}occurrence\_sim(t_i,t_j)$;
10: if $s >= \tau$ then
11: $G[i][j] = 1$; // calling the diversity metric
12: else
13: $G[i][j] = 0$;
14: return G

After completing the construction of the semantic similarity graph, the diversified coverage problem is eventually converted to the problem of finding the minimum independent dominating set (MIDS) in the semantic similarity graph. Next, we propose an exact algorithm, namely EDC, and a greedy algorithm, namely GDC, to efficiently find the MIDSs from the semantic similarity graph.

4.2. The EDC Algorithm

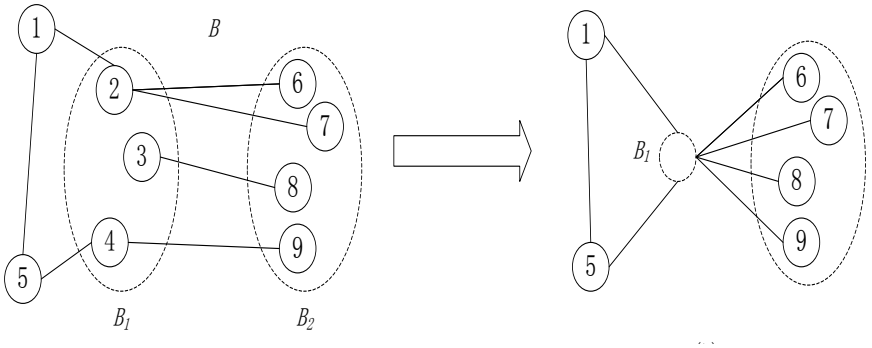
After the construction of the semantic similarity graph, the main problem is to determine the diversified coverage tag set. As ever mentioned, the diversified coverage problem is equivalent to the problem of finding the minimum independent dominating set in the semantic similarity graph. Thus, we develop an efficiency algorithm EDC to find the MIDSs from the semantic similarity graph.

Given a graph $G=\{V,E\}$, $\forall S \subseteq V$, $N(S)$ denotes the nodes in $V \backslash S$, which are the neighbors of S. Moreover, we denote $N(S) \cup S$ as $N[S]$. If C is a subset of V and any two nodes in C are linked with each other, we call C a clique. If the set V of vertices can be divided into two disjoint subsets V_1 and V_2 such that $V=V_1 \cup V_2$ and $V_1 \cap V_2=\varnothing$, and the two vertices of any edge are in V_1 and V_2, respectively, V is referred to as a bipartite graph. Given a graph $G=\{V,E\}$ and a subset $S \subseteq V$, $G[S]=\{S,E'\}$, where E' is the set of edges, in each of which the two ends are both in S.



(a) (b)

Fig. (1). Clique.



(a) (b)

Fig. (2). Bipartite.

Since the existing solutions to the MIDS problem often need to traverse the power set of all vertices in V, the time complexity is prohibitively large. In this paper, we propose a heuristic algorithm based on the following observations:

(1) Let $C \subseteq G$ be a clique in a graph G, as shown in Fig. (**1a**). Since any two nodes in C are connected with each, we have reason to consider all the nodes in C are similar. In this case, any node in C can be used as a representation of the other nodes in C. For a node outside C but connected with the nodes in C, we can compute the similarity between it and the node within C in the following way. Since every word may be of several different semantics, we compute the similarity of the two tags in a following way. First, for every pair of semantics of the two tags, the corresponding similarity is computed. Then, derive the average of all the similarities obtained as the similarity of the two tags. In this paper, we consider that the similarity is of transitivity, but it becomes smaller with the distance of two nodes lengthening. We utilize any node inside a clique to represent the clique such that any two nodes directly connected to the clique are spaced out by just one node. This measures up the intuition. Therefore, the structure in Fig. (**1a**) is converted to that in Fig. (**1b**). In this case, any node in C is selected as a representative and put in the minimum independent dominating set while we do the following operations:

$$del\_C(V, N[C]) = \{V' : V \setminus N[C]\} \tag{10}$$

(2) If there is a bipartite $B\{B_1, B_2\}$ in G, as shown in Fig. (**2**), we can use any independent part of B, i.e. B_1 or B_2, as an independent dominating set. Node 1 and node 5 are respectively connected with the nodes in B_1. If B_1 is added to the minimum independent dominating set, nodes 2 and 4 are considered to dominate nodes 1 and 5, respectively. As such, Fig. (**2a**) is converted into Fig. (**2b**). In this case, B_1 is added into the minimal independent dominating set while we do the following operation:

$$del\_B(V, N[B_1]) = \{V' : V \setminus N[B_1]\} \tag{11}$$

(3) It is not necessary that the nodes of the largest degree must be in the minimum independent dominating set. As shown in Fig. (**3**), nodes 1 and 4 consist of a minimum independent dominating set, but the degree of node 1 is just 2. This indicates that the node of the largest degree may be not the best in the selection of the minimum independent dominating set. Instead, we should consider the problem from two aspects, i.e. the nodes of the largest degree and the nodes of degree 1 or 2.

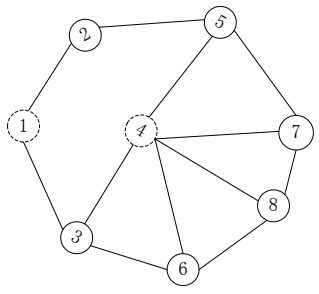


Fig. (3). The degree of nodes.

Based on the above three observations, we propose the EDC algorithm to address the problem of diversified coverage based tag recommendation. The pseudo code is given in Algorithm **4**.

Algorithm 4: EDC( )

Input: the semantic similarity graph $G=\{V,E\}$
Output: the minimum independent dominating set R'
1: $R'\leftarrow\varnothing$
2: if $|V|=0$ then
3: return $\varnothing$
4: else if $\exists C\subseteq V$ s.t. C is a clique then
5: select a random $v\in C$
6: $R'=R'\cup\{v\}$
7: d*el_C*(V,N[C])
8: else if $\exists B\subseteq V$ s.t. B is a bipartite then
9: $b=\min\{B_1,B_2\}$
10: $R'=R'\cup b$
11: d*el_B*(V,N[b])
12: else if $\exists u\in V$ s.t. degree(u)=1 or degree(u)=2 then
13: $R'=R'\cup\min\{|\text{EDC}(G[V\backslash\text{N}[u]])|, |\text{EDC}(G[V\backslash\{u\}])|\}$
14: else
15: select u whose degree is maximum
16: $R'=R'\cup\min\{|\text{EDC}(G[V\backslash\text{N}[u]])|, |\text{EDC}(G[V\backslash\{u\}])|\}$
17: return R'

In the EDG Algorithm, R' is first initialized as empty, and then the validity of data is checked. If V is empty, the algorithm returns null. Line 4-7 decide whether there exists a clique in the graph. If it exists, a node in the clique is selected randomly and added into R' while the clique and its neighbors are deleted. In Line 8-11, we decide whether there is a bipartite in the graph. If yes, the smaller part of the bipartite is selected and added into R' while the smaller part and its neighbors are deleted. If there are no such two structures in the graph, we recursively search the minimum independent dominating set in the remaining graph. Line 12 decides whether there are nodes, the degree of which are 1 or 2, in the graph. If yes, select the minimum independent dominating set in two ways. That is, delete all the nodes just of degree 1 or 2 and recursively perform in the remaining graph or delete the nodes just of degree 1 or 2 and their neighbors while recursively searching in the remaining graph. As such, the EDC algorithm traverses the entire searching space to find the minimum independent dominating set. If there is no node of degree 1 or 2 in the graph, select the node of the largest degree in Line 15-16 and repeat the above similar process.

4.3. The GDC Algorithm

EDC is an exact algorithm for the MIDS problem. However, it is of high time complexity. Thus, when handling the large-scale data, the response time will be insufferably long. In this section, we design a greedy algorithm, namely GDC, to address the MIDS prolem, where the problem is further converted to the set cover problem. Let U be a set and S the set of all non-empty subsets of U s.t. $U=\bigcup_{s\in S}s$. The minimum set cover prolem is to find a set C such that $U=\bigcup_{s\in S}s$ and $|C|$ is minimum.

Algorithm 5: GDC()

Input: the semantic similarity graph $G=\{V,E\}$
Output: the minimum independent dominating set R'
1: $S_p\leftarrow\varnothing$, $R'\leftarrow\varnothing$;
2: for each v in V do //construct $N[v]$
3: $S=S\cup N[v]$;
4: sort() //sort in the non-increasing order of $N[v]$|
5: while $\{N[v]|v\in R'\}$ cannot cover G do
6: select the vertex of the largest |N[v]|;
7: R′=R′ ∪ {v};
8: *del_s*(S,N[v]); //delete the covered vertices
9: *del_p*(S,N[v]);
10: return R′;

In a graph $G=\{V, E\}$, let $N(v)$ be all the neighbors of node v. Then, $N[v]=N(v)\cup v$ and $V=\bigcup_{v\in V}N[v]$. As such, the problem of finding the MIDSs from the semantic similarity graph can be converted to the set cover problem, where V corresponds to U and $\cup N[v]$ corresponds to S. $N[v]$ can be considered as the set of vertices dominated by v, also denoted as s_v.

The basic idea of GDC can be briefed as follows. Select node v such that $|N[v]|$ is maximum and, for any neighbor u of node v, conduct that (1) delete all nodes in s_u from S, as shown in Eq. (12), since u is covered by v and thus could not be in any MIDS; (2) likewise, delete u from the neighborhood of every node besides v, as shown in Eq.(13).

$$del\_s(S,N[v])=\{S:S=S\backslash s_u,u\in N(v)\} \quad (12)$$

$$del\_p(S,N[v])=\{s'\neq\varnothing:s'=s\backslash N[v],s\in S\} \quad (13)$$

Further, we find that the MIDSs in S are of the following properties: (1) for $s\in S$, if $\exists r\in S$ s.t. $r\neq s$ and $s\subseteq r$, there must an MIDS not containing s; (2) for $u\in S$, if $\exists s\in S$ s.t. $u\subseteq s$ and $\forall v\in S(v\neq s)$, $u\not\subseteq v$, u must be in any MIDS. By exploiting the two properties, some special cases can be efficiently handled.

The pseudo code of the GDC algorithm is given in Algorithm **5**, which is a greedy algorithm. In this algorithm, the vertices of the large $|N[v]|$, i.e. those of more neighbors, are first iterately selected. After these vertices are removed from the semantic similarity graph, the number of the remaining vertices could be greatly reduced since the neighbors of these

Table 4. An overview of two real datasets.

	Last.fm	MovieLens
# users	1892	2133
# items	17632	10197
# tags	11946	13222
# tagging records	186479	47899
# user tags	98.562	22.696
# item tags	14.891	8.117
{user, item}	92834	855598

vertices have been covered and thus not to be considered. In Line 2~4, construct a set for every vertex in the graph, and sort these sets according to their size. In the loop of Line 5~10, we decide whether the current set R can cover the graph G. If yes, the algorithm terminates; otherwise, select the vertex v of the currently largest degree and add it to R'. Then, delete the corresponding set $N[v]$ from S, i.e. $del\_s(S, N[v])$. Moreover, due to all the vertices connected to v have been covered by v, we should delete these vertices from the other sets in S, i.e. $del\_p(S, N[v])$. Next, we decide if all the vertices in G have been covered. If no, repeat the process until all the vertices are covered.

5. EXPERIMENTS

All experiments are conducted on a 2.0-GHz HP PC with 1G memory running Window XP. Both real and synthetic datasets are used in the experiments.

Two real data sets issued by the second HetRec (International Workshop on Information Heterogeneity and Fusion in Recommender Systems), i.e. Last.fm and MovieLens, are introduced for testing. They are both constructed based on the social networks, which allows users to label and share their resources freely. Table **4** gives a summarization of the two data sets.

The synthetic datasets are generated as follows. First, all tags are numbered in order. Then, with the normal distribution assumption, the numbered tags are allocated to every user such that every user is of the same number of tags. Unless otherwise specified, the default setting is *#users*=1000, *#tags*=500, K=10 and τ=0.8

5.1. Efficiency

In this set of experiments, we test the algorithm efficiency using the synthetic datasets.

By Fig. (**4a**), we study how response time varies with respect to *#users*, where *#users* are set to 500, 1000, 1500 and 2000, respectively. Fig. (**4a**) shows that the response time roughly increases in a linear way with the *#users* increasing. This is because that both EDC and GDC compute the recommendation result for every user based on the <user, tag> model. The increasing of *#users* just increases the number of repeated computations. By Fig. (**4b**), we study how response time varies with respect to *#tags*, where *#tags* are set to 500, 1000, 1500 and 2000, respectively. In Fig. (**4b**), the response time increasing is not linear but exponential. This is because that the increasing of *#tags* leads to $O(n^2)$ increasing in the co-occurrence matrix scale such that response time for the diversified coverage search exponentially increases.

By Fig. (**1c-d**), we study how response time varies with respect to K and τ. As the figures show, the response time of the EDC algorithm is much slower that that of the GDC algorithm. However, the response time is basically invariant within EDC or GDC. This shows that the two algorithms is well scalable w.r.t K and τ.

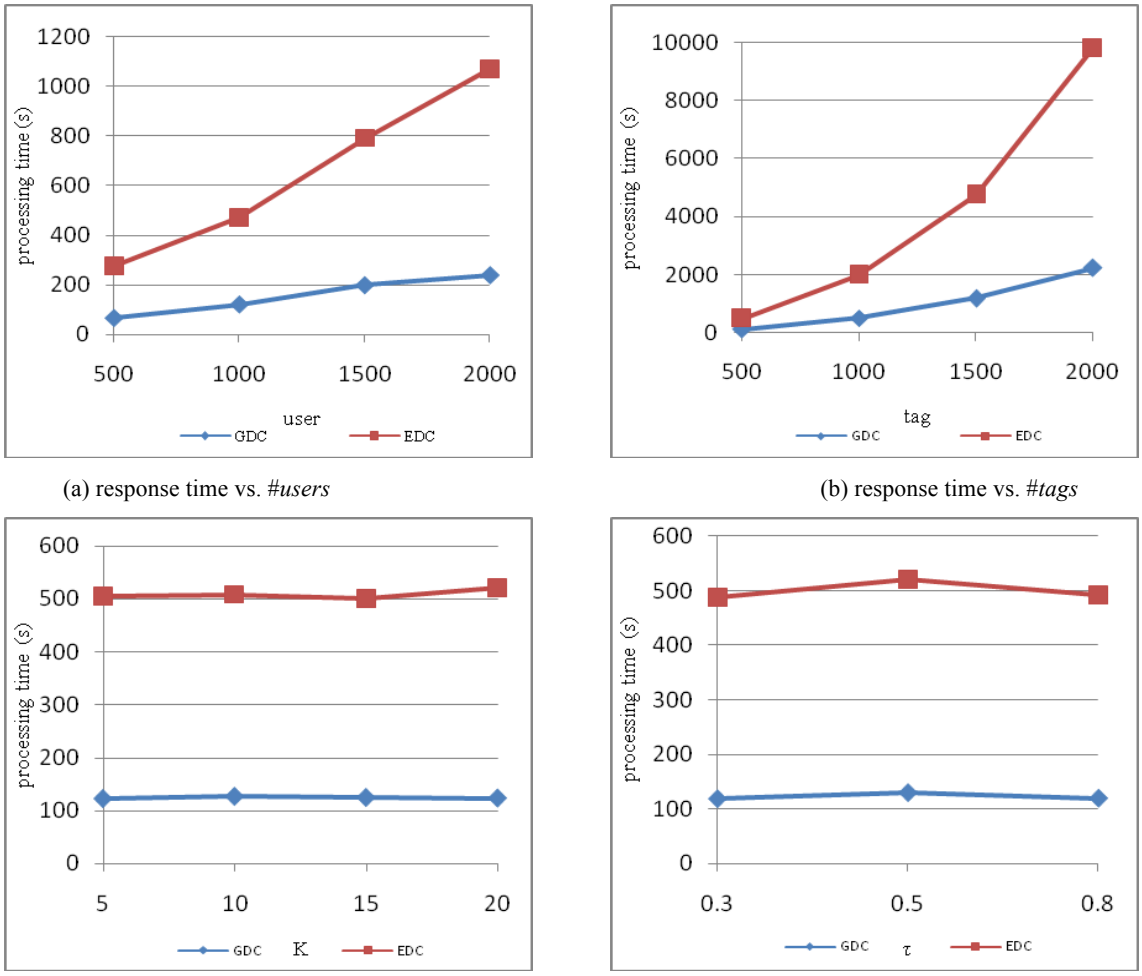
5.2. The Comparison of Similarity Measurement

In order to conduct the comparison of different similarity measurements, we introduce Resnik, Wu and Lin measurement methods [6-12] in this set of experiments. To ensure the experiment persuasiveness, the results are compared with that of Miller and Charles [7]. In the experiments of Miller and Charles, 38 undergraduates participated in the experiment, and everyone is given 30 words. The participations were requested to give the similarity between each pair of words according to their own understanding, and the similarity value ranges from 0 to 4. The evaluation from 38 undergraduates are averaged as the final experimental results.

Moreover, we adopt a third-party software, which is coded by Siddharth Patwardhan and Ted Pederson, to ensure the experimental fairness. The Software package includes a series of measurement methods, such as Leacock [8], Jiang Conrath [9], Resnik [10], Hirst St Onge [11], Wu Palmer [12], and Banerjee [13], etc., which are all based on WordNet. In this test, we adopt the latest version based on WordNet 3.0.

$$\rho = \frac{Cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}} \tag{14}$$

Table **5** shows the similarity values for every pair of words calculated by various measurement methods and their



(a) response time vs. *#users*

(b) response time vs. *#tags*

(c) response time vs. *K*

(d) response time vs. τ

Fig. (4). Efficiency.

Table 5. Semantic similarity.

Algorithm		Miller	Wu	Resink	Lin	sim
car	automobile	3.92	0.89	6.11	1.00	1.00
gem	jewel	3.84	0.86	10.52	1.00	1.00
journey	voyage	3.84	0.92	5.82	0.69	0.88
boy	lad	3.76	0.80	7.57	0.82	0.88
coast	shore	3.70	0.91	8.93	0.97	0.99
asylum	madhouse	3.61	0.82	11.50	0.98	0.97
magician	wizard	3.50	0.80	11.91	1.00	1.00
midday	noon	3.42	0.88	10.40	1.00	1.00
furnace	stove	3.11	0.46	2.56	0.22	0.39
food	fruit	3.08	0.22	0.86	0.13	0.63

Table 5. contd…

Algorithm		Miller	Wu	Resink	Lin	sim
bird	cock	3.05	0.94	7.74	0.80	0.73
bird	crane	2.97	0.84	7.74	0.67	0.73
tool	implement	2.95	0.91	7.10	0.92	0.97
brother	monk	2.82	0.92	10.99	0.25	0.33
crane	implement	1.68	0.67	3.74	0.39	0.59
lad	brother	1.66	0.60	2.54	0.29	0.28
journey	car	1.16	0.00	0.00	0.00	0.00
monk	oracle	1.10	0.46	2.54	0.23	0.34
cemetery	woodland	0.95	0.18	0.86	0.08	0.19
food	rooster	0.89	0.13	0.86	0.10	0.40
coast	hill	0.87	0.67	6.57	0.71	0.71
forest	graveyard	0.84	0.18	0.86	0.00	0.19
shore	woodland	0.63	0.44	1.37	0.14	0.30
monk	slave	0.55	0.60	2.54	0.25	0.39
coast	forest	0.42	0.40	1.37	0.13	0.29
lad	wizard	0.42	0.60	2.54	0.27	0.32
chord	smile	0.13	0.44	2.80	0.27	0.35
glass	magician	0.11	0.36	2.50	0.13	0.31
noon	string	0.08	0.00	0.00	0.00	0.00
rooster	voyage	0.08	0.00	0.00	0.00	0.00
correlation coefficient		1.00	0.74	0.77	0.80	0.84

correlation coefficients with Miller method. Concretely, the correlation coefficient is computed according to Eq. (14). Since the methods are based on different measurements, not all of the similarity values are within 0 to 1. From Table **5**, we draw a conclusion as follows. In the two WordNet-based semantic similarity measurements, i.e. pathway based semantic similarity and IC based semantic similarity, the latter is better. This is because that the pathway based semantic similarity just takes into account the paths between two concepts while ignoring the IC value of every concept. However, IC is the soul of a concept, which determines the conceptual semantics.

5.3. Correlation and Diversity

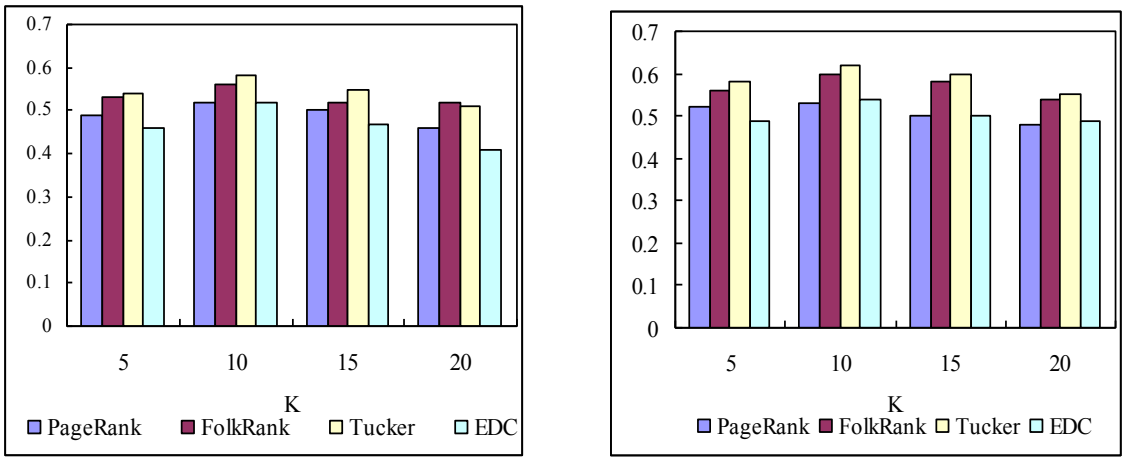
In this set of experiments, two common used measurements are introduced to study the algorithm effectiveness.

Correlation. nDCG (Normalized Discounted Cumulative Gain) is utilized to measure the tag diversity, which is often used in search engine. *nDCG* considers not only the correlation, but the influence of position to the recommendation quality. In this method, every document has a contribution to its position, and the contribution value is associated with the document correlation. Therefore, *nDCG* is formalized as follows:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{15}$$

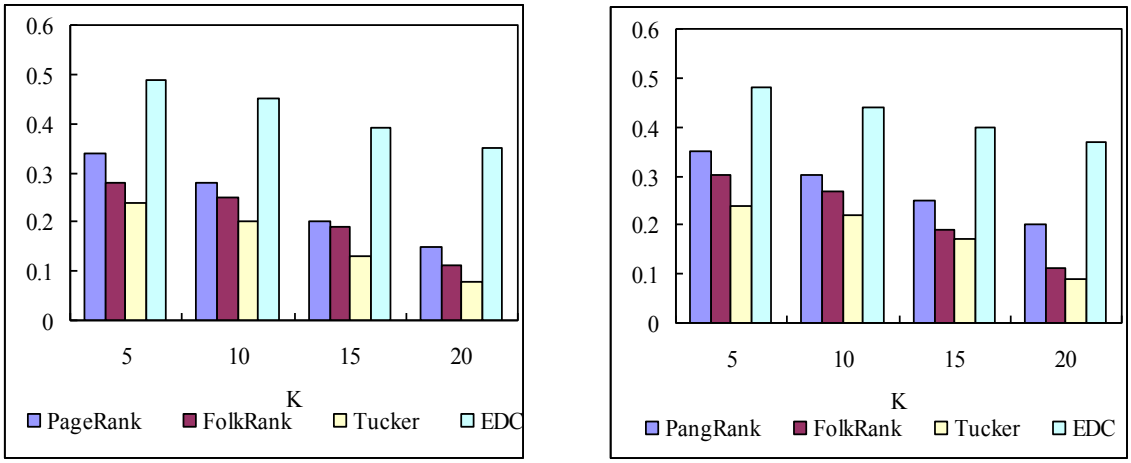
$$\text{Where } DCG_p = \sum_{i=1}^{p} \frac{r_i}{\log_2(i+1)} \tag{16}$$

In Eq. (16), r_i is the correlation of the i^{th} tag in the recommendation result set. The more the tag is at the back, the contribution of the tag is smaller. To punish the tags ranked behind but of high correlation, we devide the *log* value of the number of positions. As such, the accumulated *DCG* values of top-*p* tags is given by Eq. (15). $IDCG_p$ in Eq.(15) is the



(a) Last.fm (b) MovieLens

Fig. (5). *nDCG* vs. *K*.



(a) Last.fm (b) MovieLens

Fig. (6). *AILD* vs. *K*.

ideal *DCG* value. Sorting the recommendation results according to the correlation degree, and computing the DCG_p value, we obtain $IDCG_p$.

Diversity. We adopt the AILD (Average IntraList Distance) to measure the tag diversity, which can be computed as follows::

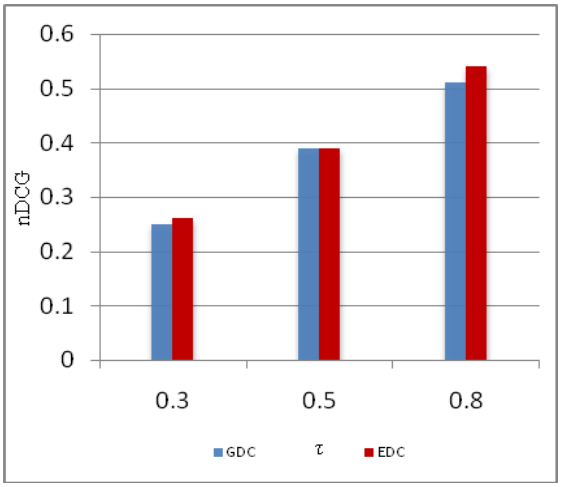
$$AILD_p = \frac{1}{K'} \sum_{i=1}^{p} \sum_{j=i+1}^{p} dist(t_i, t_j) \tag{17}$$

where $K'=(k^2-k)/2$, $dist(t_i,t_j)=1-sim(t_i,t_j)$. The larger the AILD of the recommendation result set, the less the similarity is; otherwise, the less the similarity is.

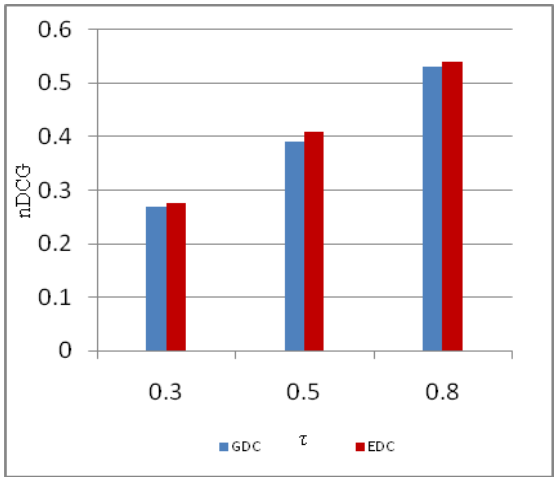
Specially, PageRank [14] and its persionalized improvements, i.e. FolkRank [15] and Tucker [16] are introduced as the compared algorithms.

Fig. (**5** and **6**) show how *nDCG* and *AILD* vary with *K* increasing in *Last.fm* and *MovieLens* datasets. As seen from the two figures, we see that, compared with the other algorithms, the *nDCG* value of the EDC algorithm is lower, but the *AILD* value is higher. This is because the traditional algorithms do not consider the tag diversity such that there are many redundant but highly similar results existing in the result set. The case can be considered as the results of the same semantic occur twice or several times in the result set. Since the correlation is increased, the *nDCG* value is large. However, the correlation is not the unique criterion. The GDC and the EDC algorithms take into account the tag diversity. Thus, many redundant results are removed and replaced by the ones of less similarity. As such, the diversity of the result set is increased. As seen from the figures, the EDC and the GDC algorithms behave far better than the alternative algorithms.

Additionally, with *K* increasing, the difference of the *nDCG* values between the EDC and the GDC algorithms becomes smaller and smaller. This is because the result sets generated by the EDC and the GDC algorithms become more and more similar with *K* increasing. Further, more and

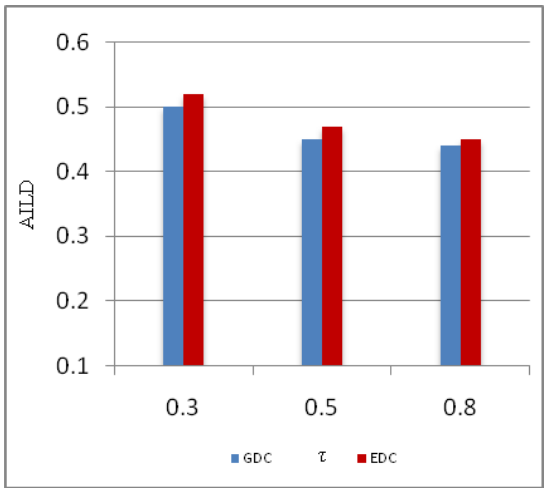


(a) Last.fm

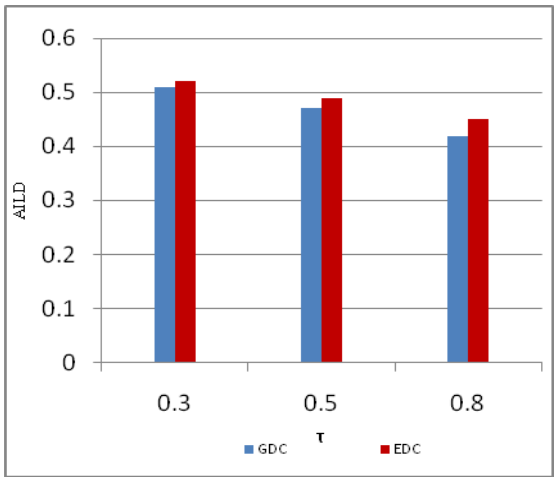


(b) MovieLens

Fig. (7). *nDCG* vs. *τ*.



(a) Last.fm



(b) MovieLens

Fig. (8). *AILD* vs. *τ*.

more vertices of less tag correlation are selected and added to the result set with *K* increasing. Thus, the AILD value reduces to some extent.

Fig. (**7** and **8**) show how *nDCG* and *AILD* vary with *τ* increasing in *Last.fm* and *MovieLens* datasets. Since there is no *τ* in the other algorithms, we only compare the EDC and the GDC algorithms. When *τ* is small, the semantic similarity graph is dense. Since the vertex degree becomes larger, the recommendation result generated in this case are often the vertices of the larger degree but not necessarily that of the larger similarity. This leads to the smaller *nDCG* values. With *τ* increasing, the graph is sparser and sparser such that more vertices of the large tag correlation are added to the result set. This leads to the large *nDCG* value. However, from the perspective of AILD, the similarity between the vertices directly connected is very small because of the small *τ*. The large distance between the vertices not directly connected leads to the large AILD of the result set. With *τ* increasing, this case varies. That is, the AILD value becomes smaller.

CONCLUSION

With the development of Web2.0, the Internet strides into a data explosion era. The appearing of the recommender engine has brought a more personalized experience for the users, relieving the users from the massive data. In this paper, based on the previous work, we study the diversified coverage problem in tag recommendation in order to improve the quality of recommendation results.

In this paper, we design a novel correlation measure based on the local and the global tag co-occurrence matrices. By integrating the users' personal interests and the degree of tag recognition, the accuracy of recommendation is

improved; In diversity, using the WordNet dictionary, we define the semantic diversity from the perspective of IC such that the semantic redundancy in the results was effectively alleviated; Further, by mapping diversity based tag coverage problem to the MIDS problem, an exact algorithm, namely EDC, and a greedy algorithm, namely GDC, are proposed; In the experiments, we verify the efficiency and effectiveness of the proposed algorithms based on two real datasets, i.e. MovieLens and Last.fm. The experimantal analysis are conducted in terms of both *nDCG* and AILD. The results show that our methods significantly improves the diversity of the results.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

This work was supported by National Natural Science Foundation of China (61272182, 61100028, 61073063, 61173030), 863 program (2012AA011004), 973 program (2011CB302200-G), National Science Fund for Distinguished Young Scholars (61025007), State Key Program of National Natural Science of China (61332014), New Century Excellent Talents (NCET-11-0085), Fundamental Research Funds for the Central Universities (N130504001), and China Postdoctoral Science Foundation (2012T50263, 2011M500568).

REFERENCES

[1] M. Costa, D. Gomes, F. Couto, and M. Silva, "A survey of web archive search architectures", In: *Proceedings of the 22nd International Conference on World Wide Web Companion (WWW '13 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2013, pp. 1045-1050.

[2] X. Tang, M. Zhang and C.C. Yang, "*User Interest and Topic Detection for Personalized Recommendation*", Web Intelligence, pp. 442-446, 2012.

[3] A. Benlian, R. Titah and T. Hess, "Differential effects of provider recommendations and consumer reviews in e-commerce transactions: an experimental study", *Journal of Management Information Systems*, vol. 29, no. 1: pp. 237-272, 2012.

[4] F.A. Durão and P. Dolog, "A Personalized Tag-Based Recommendation in Social Web Systems", CoRR abs/1203.0332, 2012.

[5] Cantador, I. Konstas and J.M. Jose, "Categorising social tags to improve folksonomy-based recommendations, Web Semantics", Science, *Services and Agents on the World Wide Web*, vol. 9, no. 1, pp. 1-15, 2011.

[6] D. Lin, "An information-theoretic definition of similarity", In: *Proceedings of the ICML*, 1998, pp. 296-304.

[7] G.A. Miller and W.G. Charles, "Contextual correlates of semantic similarity", *Language and Cognitive Processes*, vol. 6, no. 1, pp. 1-28, 1991.

[8] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification", *WordNet: An electronic lexical database*, vol. 49, no. 2, pp. 265-283, 1998.

[9] J.J. Jiang and D.W. Conrath, "*Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy*", arXiv preprint cmp-lg/9709008, 1997.

[10] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy", *IJCAI*, pp. 448-453, 1995.

[11] G. Hirst, and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms", *WordNet: An electronic lexical database*, vol. 305, pp. 305-332, 1998.

[12] Z. Wu and M. Palmer, "Verbs semantics and lexical selection", In: *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, 1994, pp. 133-138.

[13] S. Banerjee and T. Pedersen, "Extended gloss overlaps as a measure of semantic relatedness", In: *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI)*, 2003, pp. 805-810.

[14] L. Page, S. Brin, R. Motwani and T. Winograd, "*The PageRank citation ranking: Bringing order to the web*", 1999.

[15] R. Schifanella, A. Barrat, C. Cattuto, B. Markines, and F. Menczer. "Folks in folksonomies: social link prediction from shared metadata", In: *Proceedins of the 3rd ACM International Conference on Web Search and Data Mining*, 2010, pp. 271-280.

[16] S. Rendle, L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation", In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, 2010, pp. 81-90.

Received: September 22, 2014 Revised: November 03, 2014 Accepted: November 06, 2014