

# A Graph-based Recommendation across Heterogeneous Domains

Deqing Yang<sup>§\*</sup>, Jingrui He<sup>‡</sup>, Huazheng Qin<sup>§</sup>, Yanghua Xiao<sup>§</sup>, Wei Wang<sup>§</sup>  
<sup>§</sup>School of Computer Science, Shanghai Key Laboratory of Data Science  
<sup>§</sup>Fudan University, Shanghai, 200433, China P. R.  
<sup>§</sup>{yangdeqing, huazhengqin13, shawyh, weiwang1}@fudan.edu.cn  
Arizona State University, AZ, USA, <sup>‡</sup>jingrui.he@gmail.com

## ABSTRACT

Given the users from a social network site, who have been tagged with a set of terms, how can we recommend the movies tagged with a completely different set of terms hosted by another website? Given the users from a website dedicated to Type I and Type II diabetes, how can we recommend the discussion threads from another website dedicated to gestational diabetes, where the keywords used in the two websites might be quite diverse? In other words, how can we recommend across heterogeneous domains characterized by barely overlapping feature sets?

Despite the vast amount of existing work devoted to recommendation within homogeneous domains (e.g., with the same set of features), or collaborative filtering, emerging applications call for new techniques to address the problem of recommendation across heterogeneous domains, such as recommending movies hosted by one website to users from another website with barely overlapping tags. To this end, in this paper, we propose a graph-based approach for recommendation across heterogeneous domains. Specifically, for each domain, we use a bipartite graph to represent the relationships between its entities and features. Furthermore, to bridge the gap among multiple heterogeneous domains with barely overlapping sets of features, we propose to infer their semantic relatedness through concept-based interpretation distilled from online encyclopedias, e.g., Wikipedia and Baike. Finally, we propose an efficient propagation algorithm to obtain the similarity between entities from heterogeneous domains. Experimental results on both Weibo-Douban data set and Diabetes data set demonstrate the effectiveness and efficiency of our algorithm.

## Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: On-line Information Services-Web-based services; H.2.8 [Database Management]: Database applications-Data mining; I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods-Semantic networks

\*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).  
*CIKM'15*, October 19-23, 2015, Melbourne, VIC, Australia  
© 2015 ACM. ISBN 978-1-4503-3794-6/15/10 ...\$15.00.  
DOI: <http://dx.doi.org/10.1145/2806416.2806523>.

## Keywords

cross-domain recommendation, heterogenous domains, graph propagation, semantic matching

## 1. INTRODUCTION

Up until now, recommender systems have been successfully applied to a variety of domains, ranging from movies to research papers, from online friends to social tags. Existing techniques for recommender systems can be mainly categorized into three different groups: *collaborative filtering* (*CF for short*) [15, 16], *content-based filtering* [3, 6], and *hybrid recommender systems* [5]. Most of these techniques focus on a single domain [27, 36]. In other words, the users and the items come from the same website.

More recently, researchers have studied cross-domain recommendation [19, 21, 24, 32], where the users and the items come from different websites. The main goal of cross-domain recommendation is to alleviate the cold start problem, such that the matching between users and items across different domains is satisfactory in the beginning. Notice that, most, if not all, of existing techniques for cross-domain recommendation work best if different domains share a large number of features. In this way, the relevance between users from one domain and items from another can be naturally obtained based on the common features. Please refer to Section 4 for a detailed review of related works.

### 1.1 Applications

Nowadays, with the emergence of various e-commerce websites, a natural question is: how can we recommend items across *heterogeneous* domains, i.e., the different domains sharing very few features? Take Weibo (Chinese version of Twitter, <http://www.weibo.com>) and Douban (the largest Chinese movie database, <http://www.douban.com>) as an example. The tags used by Weibo users are quite different from those used to specify movies on Douban, which is illustrated by the tag clouds in Figure 1. From this figure, we can see that the common Weibo tags include 'music', 'movie', 'food', '80s', which generally characterize a Weibo user's hobby, age, constellation, etc. In contrast, Douban tags are generally used to specify a movie's nation, year and genre, such as 'USA', 'Japan', 'animation', 'classic', etc. For this case, existing techniques for cross-domain recommendation may not be able to accurately identify the movies from Douban that may be of potential interest to Weibo users. Another example is the recommendation of articles across different websites. In particular, among the various social network websites dedicated to diabetes patients, some of them might be open to all types of diabetes patients, such as <http://www.tudiabetes.org> (denoted by



chase exist in each domain. However, all the solutions for these cases obviously can not be directly used to solve the problem setting as shown in Figure 2(e), where domain A only contains users and domain B only contains items and no shared features can be used to establish the links between the users in domain A and the items in domain B.

However, the heterogenous cross-domain recommendation scenario of Figure 2(e) is a popular and valuable setting in real world. For example, it will create a lot of business opportunities if we can accurately recommend products in eBay to a user in Tweet or Facebook. Therefore, in this paper, we focus on the challenging problem of recommendation across heterogeneous domains. In general, this new problem setting poses two challenges.

1. The first challenge is the sparsity of shared features across heterogeneous domains, which makes the discovery of the connections among the non-shared features non-trivial. Recall the tags of Weibo users and Douban movies, correlating different tags across the two domains is critical for recommending Douban movie to Weibo users. However, discovering such connections is not a trivial task.
2. The second challenge is designing an effective and efficient model for inferring the similarity of entities across heterogeneous domains, especially when the numbers of entities and features are very large. Recall the setting of Figure 2(e), even when the connections between the features of the two domains are discovered, inferring the similarity between users and items is also challenging.

### 1.3 Basic Idea and Organization

To overcome these challenges, we propose to use bipartite graphs to represent the relationships between entities and their features, and build the connection between heterogeneous domains through semantic matching of features. Based on the composite multi-partite graph, we propose an effective and efficient propagation algorithm to infer the global similarity between entities across different domains. The major contributions of this paper can be summarized as follows.

1. We propose a novel recommendation approach based on a multi-partite graph representation and an efficient propagation algorithm. Furthermore, we justify the optimization of our algorithm.
2. We conduct extensive experiments on two real applications: Douban movie recommendation for Weibo users, article recommendation across different diabetes social networks, to verify the effectiveness of our solution for recommendation across heterogeneous domains.

The rest of this paper is organized as follows. In Section 2, we introduce our graph-based approach for recommendation across heterogeneous domains including the semantic matching between different tags/keywords. In Section 3, we evaluate the performance of our approach against existing techniques. Finally, we survey the related works in Section 4 and conclude our paper in Section 5.

## 2. GRAPH-BASED APPROACH FOR CROSS-DOMAIN RECOMMENDATION

Different from traditional CF-based methods, in this paper, we target a more challenging problem of cross-domain recommendation, i.e., recommending the items from one domain to the users from another heterogeneous domain. This problem essentially boils

down to inferring the similarity between different types of objects (users and items) from multiple domains. In this section, we present our graph-based method: we first start with some notations, followed by the introduction of the global similarity and the efficient computation of the relevance vectors, and finally discuss semantic matching for inducing cross-domain correlation.

### 2.1 Notation

Without loss of generality, in this paper, we consider 2 different domains, although the proposed method can be easily generalized to multiple domains. Formally, for the  $i^{\text{th}}$  domain ( $i=1,2$ ), we use a bi-partite graph  $G_i = \{V_i, E_i\}$  to represent the relationships between the objects in this domain and their characterizing features, where  $V_i$  is the set of nodes in this graph, and  $E_i$  is a set of undirected edges. Let  $n_i$  denote the number of objects in the  $i^{\text{th}}$  domain, and  $m_i$  denote the number of features. Therefore,  $V_i$  consists of two types of nodes:  $n_i$  object nodes, and  $m_i$  feature nodes. Notice that in  $E_i$ , an edge only exists between different types of nodes, i.e., between an object node and a feature node, and the edge weights are set to the feature value of the corresponding object. Let  $\mathbf{X}_i$ ,  $n_i \times m_i$ , denote the connectivity matrix between the two types of nodes, whose elements are set to be the edge weights.

In cross-domain recommendation where only a small fraction of features are shared across different domains, a critical question is how to build the connection between the bi-partite graphs for different domains. To address this problem, in our method, we propose the following matching graph between features from the two domains. Let  $G_0 = \{V_0, E_0\}$  denote such a bi-partite matching graph, where  $V_0$  includes all the feature nodes from the two domains, and  $E_0$  denotes the set of edges connecting features from different domains. Based on this graph, we define connectivity matrix  $\mathbf{X}_0$ ,  $(m_1 + m_2) \times (m_1 + m_2)$ , whose elements measure the similarity between features from different domains. Details of learning the connectivity matrix  $\mathbf{X}_0$  based on semantic matching will be discussed in Subsec. 2.4.

Putting all above graphs together, we get a multi-partite graph  $G = \{V, E\}$  as shown in Figure 3, where  $V = V_1 \cup V_2$ , and  $E = E_1 \cup E_2 \cup E_0$ . Then, we define an *affinity matrix*  $\mathbf{X}$  for this graph.  $\mathbf{X}$  is an  $(n_1 + n_2 + m_1 + m_2) \times (n_1 + n_2 + m_1 + m_2)$  matrix and is represented as,

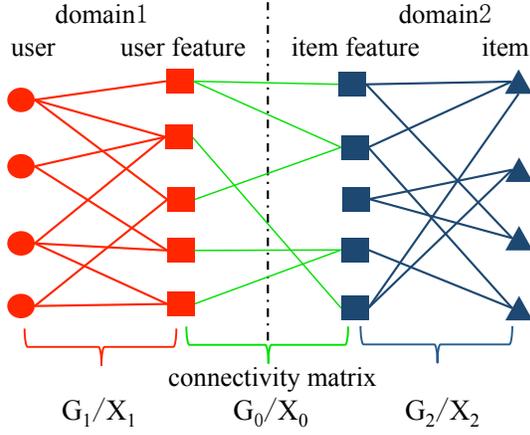
$$\mathbf{X} = \begin{bmatrix} \mathbf{0}_{n_1 \times n_1} & \mathbf{0}_{n_1 \times n_2} & \mathbf{X}_1 & \mathbf{0}_{n_1 \times m_2} \\ \mathbf{0}_{n_2 \times n_1} & \mathbf{0}_{n_2 \times n_2} & \mathbf{0}_{n_2 \times m_1} & \mathbf{X}_2 \\ \mathbf{X}_1^T & \mathbf{0}_{m_1 \times n_2} & \mathbf{0}_{m_1 \times m_1} & \mathbf{X}_0 \\ \mathbf{0}_{m_2 \times n_1} & \mathbf{X}_2^T & \mathbf{X}_0^T & \mathbf{0}_{m_2 \times m_2} \end{bmatrix}$$

where  $(\cdot)^T$  denotes matrix transpose and  $\mathbf{0}$  is a zero matrix. Based on this graph, our goal is to infer the similarity between object nodes from different domains.

### 2.2 Global Similarity between Objects

In graph  $G$ , the direct connections between object nodes from different domains are absent, e.g., between user nodes and item nodes in Figure 3. Thus, we will measure their similarity based on graph propagation. To be specific, we first normalize the affinity matrix  $\mathbf{X}$  as follows.

$$\begin{aligned} \mathbf{S} &= \mathbf{D}^{-\frac{1}{2}} \mathbf{X} \mathbf{D}^{-\frac{1}{2}} \\ &= \begin{bmatrix} \mathbf{0}_{n_1 \times n_1} & \mathbf{0}_{n_1 \times n_2} & \mathbf{S}_1 & \mathbf{0}_{n_1 \times m_2} \\ \mathbf{0}_{n_2 \times n_1} & \mathbf{0}_{n_2 \times n_2} & \mathbf{0}_{n_2 \times m_1} & \mathbf{S}_2 \\ \mathbf{S}_1^T & \mathbf{0}_{m_1 \times n_2} & \mathbf{0}_{m_1 \times m_1} & \mathbf{S}_0 \\ \mathbf{0}_{m_2 \times n_1} & \mathbf{S}_2^T & \mathbf{S}_0^T & \mathbf{0}_{m_2 \times m_2} \end{bmatrix} \quad (1) \end{aligned}$$



**Figure 3: A multi-partite graph across two domains. Red lines are user-feature edges in domain 1 and blue lines are item-feature edges in domain 2. The green lines between red rectangles and blue rectangles are the edges between user features and item features, which can be established by the semantic similarity of features.**

where  $D$  is a diagonal matrix with each element equal to the row sum of  $X$ ;  $S_1$ ,  $S_2$ , and  $S_0$  are normalized versions of  $X_1$ ,  $X_2$ , and  $X_0$ , respectively.

In order to compute the global similarity between the  $i^{\text{th}}$  node and all the other nodes in the composite multi-partite graph, we use  $v_i$  to denote the  $(n_1 + n_2 + m_1 + m_2)$  dimensional vector, whose  $i^{\text{th}}$  element is 1 and all the others are 0. Using the manifold ranking algorithm proposed in [35], the global similarity vector with respect to the  $i^{\text{th}}$  node can be written as  $(I - \alpha S)^{-1} v_i$ , where  $I$  is an  $(n_1 + n_2 + m_1 + m_2) \times (n_1 + n_2 + m_1 + m_2)$  identity matrix, and  $\alpha$  is a positive scalar whose value is close to 1. Putting all these vectors together, we have the following  $(n_1 + n_2 + m_1 + m_2) \times (n_1 + n_2 + m_1 + m_2)$  global similarity matrix  $K$ ,

$$K = (I - \alpha S)^{-1} [v_1, \dots, v_{n_1+n_2+m_1+m_2}] = (I - \alpha S)^{-1} \quad (2)$$

Each element in  $K$  measures the global similarity between each pair of nodes. The property of the global similarity matrix  $K$  can be interpreted using Taylor expansion [14], i.e.,  $K = \sum_{i=0}^{\infty} (\alpha S)^i$ . Notice that the  $i^{\text{th}}$  term measures the weighted  $i^{\text{th}}$ -step similarity on the multi-partite graph, and its impact on the overall similarity decreases exponentially with respect to  $i$ .

It is easy to see that  $K$  has the following block structure.

$$K = \left( \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \right)^{-1} = \begin{bmatrix} K_1 & K_2 \\ K_2^T & K_3 \end{bmatrix}$$

According to Equation 2, we conclude that,  $A = \begin{bmatrix} I - 0_{n_1 \times n_1} & 0_{n_1 \times n_2} \\ 0_{n_2 \times n_1} & I - 0_{n_2 \times n_2} \end{bmatrix}$ ,  $B = \begin{bmatrix} -\alpha S_1 & 0_{n_1 \times m_2} \\ 0_{n_2 \times m_1} & -\alpha S_2 \end{bmatrix}$ , and  $C = \begin{bmatrix} I & -\alpha S_0 \\ -\alpha S_0^T & I \end{bmatrix}$ .  $K_1$ ,  $K_2$ , and  $K_3$  are submatrices of  $K$ , which are of the same dimensionality as  $A$ ,  $B$ , and  $C$  respectively. Since we are only interested in the similarity among objects from different domains, instead of the whole matrix  $K$ , we only compute the submatrix  $K_1$ ,  $(n_1 + n_2) \times (n_1 + n_2)$ , which has the following closed form solution.

LEMMA 1.

$$K_1 = (I - BC^{-1}B^T)^{-1} \quad (3)$$

**Proof.** Lemma 1 can be proved by applying the Woodbury formula [25] on Equation 2, and leveraging the block structure of both  $K^{-1}$  and  $K$ . ■

Notice that the inverse of both  $C$  and  $BC^{-1}B^T$  exists. This is because  $K_1$  is part of  $K$ , which is the inverse of the symmetric positive definite matrix  $I - \alpha S$ .

Based on  $K_1$ , if the objects in the first domain are users, and the objects in the second domain are items, the relevance between the  $k^{\text{th}}$  user and all the items can be obtained from the  $k^{\text{th}}$  row or column of  $K$ , since  $K_1$  is a symmetric matrix. In other words, the *relevance vector* can be written as  $s_i = K_1 u_i$ , where  $u_i$  is an  $n_1 + n_2$  dimensional vector. The elements of  $u_i$  are 0 except the  $i^{\text{th}}$  one, which is set to 1.

### 2.3 Efficient Computation of Relevance Vector

In this subsection, we focus on the efficient computation of the relevance vector, which is summarized in Algorithm 1. It takes as input matrices  $B$  and  $C$ , the query vector  $u$  (with only one non-zero entry),  $\alpha$  and the number of iterations  $n_{iter}$ . The output is the relevance vector  $s$  with respect to  $u$ . Here we define  $\tilde{S}_0 = \begin{bmatrix} 0 & S_0 \\ S_0^T & 0 \end{bmatrix}$ . In this way, matrix  $C$  has the following equivalent form  $C = I - \alpha \tilde{S}_0$ .

The algorithm works as follows. In Step 1, we initialize the relevance vector to be the query vector  $u$ . In the outer loop starting from Step 2, we set  $w$  to be  $B^T s$  in Step 3, initialize  $r$  to be  $w$  in Step 4, and pass both vectors to the inner loop starting from Step 5. In the inner loop, we update  $r$  in Step 6  $n_{iter}$  times. Finally, in the outer loop, we re-scale  $r$  in Step 8, based on which we obtain the current estimate of the relevance vector  $s$ .

---

**Algorithm 1** Graph-based Recommendation across Heterogeneous Domains

---

**Input:**  $B, C, u, \alpha, n_{iter}$

**Output:**  $s$

- 1: Initialize  $s$  to be  $u$
  - 2: **for**  $i = 1$  to  $n_{iter}$  **do**
  - 3:    $w \leftarrow B^T s$
  - 4:   Initialize  $r$  to be  $w$
  - 5:   **for**  $j = 1$  to  $n_{iter}$  **do**
  - 6:      $r \leftarrow \alpha \tilde{S}_0 r + (1 - \alpha) w$
  - 7:   **end for**
  - 8:    $r \leftarrow \frac{1}{1 - \alpha} r$
  - 9:    $s \leftarrow B r + (1 - \alpha) u$
  - 10: **end for**
- 

The convergence property of Algorithm 1 is presented in the following theorem.

**THEOREM 1.** As  $n_{iter}$  goes to infinity,  $s$  returned by Algorithm 1 converges to  $(1 - \alpha) K_1 u$ .

**Proof.** We prove Theorem 1 via the following two steps. In the first step, we analyze the convergence property of the inner loop between Step 5 and Step 7 of Algorithm 1; then in the second step, we analyze the convergence property of the outer loop between Step 2 and step 10.

First, the following lemma shows the convergence property of the inner loop.

LEMMA 2. Given  $w$ , as  $n_{iter}$  goes to infinity, in the inner loop of Algorithm 1,  $r$  converges to  $(1 - \alpha)C^{-1}w$ .

**Proof.** To prove Lemma 2, simply observe that upon convergence,  $r = \alpha\tilde{S}_0 r + (1 - \alpha)w$ . Therefore,  $(I - \alpha\tilde{S}_0)r = (1 - \alpha)w$ . Together with the fact that  $C = I - \alpha\tilde{S}_0$ , we conclude the proof for Lemma 2. ■

Based on Lemma 2, we can see that the inner loop provides an efficient way of computing  $r = (1 - \alpha)C^{-1}w$ . Combined with Step 2 and Step 8 of Algorithm 1, we can see that in Step 9, the vector  $r$  is equal to  $C^{-1}B^T s$ . Upon convergence,  $s = BC^{-1}B^T s + (1 - \alpha)u$ . Therefore,  $(I - BC^{-1}B^T)s = (1 - \alpha)u$ . Together with the definition of  $K_1$  in Equation (3), we conclude our proof for Theorem 1. ■

Comparing Algorithm 1 with direct computation of  $s = (I - BC^{-1}B^T)^{-1}u$ , the major advantage is its time efficiency. In our applications, it is often the case that both  $B$  and  $\tilde{S}_0$  are very sparse. Let  $l_B$  and  $l_{\tilde{S}_0}$  denote the number of non-zero elements in  $B$  and  $\tilde{S}_0$  respectively. The time complexity of Algorithm 1 can be shown as follows.

LEMMA 3. The time complexity of Algorithm 1 is  $O(l_B + l_{\tilde{S}_0})$ .

**Proof.** It is easy to see that the major computation in Algorithm 1 is matrix vector multiplication. Since both  $B$  and  $\tilde{S}_0$  are very sparse, the computational complexity of  $\tilde{S}_0 r$  is  $O(l_{\tilde{S}_0})$ , and the computational complexity of  $B$  is  $O(l_B)$ , which completes the proof. ■

Therefore, Algorithm 1 scales to large data sets since its time complexity is linear with respect to the number of non-zeros elements in  $B$  and  $\tilde{S}_0$ . On the other hand, for direct computation of  $s$  via matrix inversion, the typical time complexity is  $O((n_1 + n_2)^{2.373})$  using optimized CW-like algorithms [12], which is prohibitive for large data sets.

## 2.4 Semantic Matching

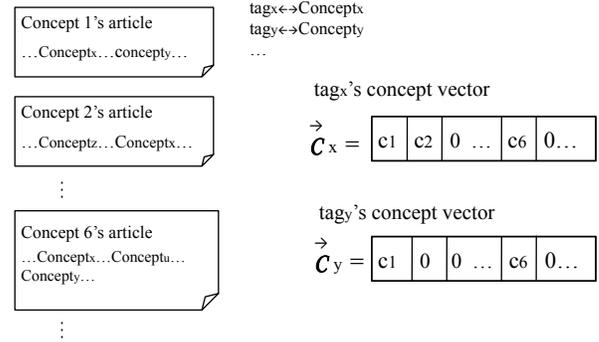
According to our algorithm, some relationships between features should be established in order to discover the similarity of objects in the two domains, i.e., inferring  $E_0$  in  $G_0$ . In this subsection, we introduce how to find the relationships between different features from two heterogeneous domains, respectively. For a better interpretation, we use the case where the features are represented by tags to illustrate our method.

### 2.4.1 Basic Idea

As we declared before, most tags coming from heterogeneous domains are different, which demands us to correlate two tags on semantic level rather than lexical level. In general, semantic matching requires more context information of tags. To accomplish it, we try to discover hyper-links existing among millions of concepts in online encyclopedias.

Most prior works on computing semantic relatedness resorted to some well-defined lexical resources, such as WordNet [18]. However, lexical resources require lexicographic expertise and only cover a small fraction of language lexicons. As we observed, the tags adopted by Weibo users contain many proper names, neologisms, and etc., which are seldom included in current lexical resources. The authors in [11, 22] used substantial semantic information of online encyclopedias, e.g., Wikipedia, to enrich the semantic representation of terms. In this paper, we also utilize online encyclopedias to achieve the semantic matching of tags. Concretely, we adopt Wikipedia and Baike (<http://baike.baidu.com>) for English tags and Chinese tags, respectively.

As we known, an online encyclopedia contains millions of concepts including person, location, organization, hobby and etc. There exists an article page describing the fact about each concept. As well, there are many hyper-links linking to other concepts on each article page to enrich its semantic description. According to the principle of previous Wikipedia based methods [11, 22], two terms, i.e., two concepts, are considered semantically related if they co-occur as hyper-links in one article page. The more such article pages can be found, the more semantically related the two terms are.



**Figure 4: The articles in online encyclopedias derive the concept vectors of tags. The left part display some articles in the online encyclopedia. In this case,  $tag_x$  and  $tag_y$  are mapped into  $Concept_x$  and  $Concept_y$  respectively, and then  $tag_x$  is semantically related to  $tag_y$  through some encyclopedia concepts although they are not represented by the same term.**

### 2.4.2 Interpreting Semantic Relationships

Then, we formalize the semantic interpretation of a tag. In order to represent the semantics of a tag through the contexts provided by the encyclopedia, we should first map a tag into an article entry (i.e., a concept) of the encyclopedia. Given a tag  $i$  and a concept  $c$ , we map  $i$  into  $c$  if  $i$  and  $c$  are exactly same or  $c$ 's string is a maximal substring of  $i$ 's string. Under this mapping scheme, we can find a mapped concepts for nearly 90% of tags. As depicted in Figure 4,  $tag_x$  and  $tag_y$  are first mapped into  $Concept_x$  and  $Concept_y$ , respectively. Then, we can find that  $Concept_x$  and  $Concept_y$  co-occur in the article page of Concept 1, indicating their semantic relatedness. Similarly, we may find another evidence, i.e., Concept 6's article page. Therefore, although  $Concept_x$  and  $Concept_y$  are not same, they are still semantically related. Clearly, the indirect matching through the contexts provided by concepts' article pages in online encyclopedia rather than lexical matching, is exactly what we expect in the semantic matching for two tags.

According to *Explicit Semantic Analysis (ESA for short)*'s basic idea [11], the semantic interpretation of a tag  $i$  can be represented by a *concept vector* which is formally defined as  $\vec{C}_i = [c_1, \dots, c_C] \in \mathbb{R}^C$ .  $C$  is the total number of concepts in the encyclopedia and  $c_j (1 \leq j \leq C)$  represents the semantic relevance of concept  $j$  to tag  $i$ , i.e., the TF-IDF score of concept  $i$ 's occurrence in concept  $j$ 's article. Figure 4 describes the compositions of  $\vec{C}_x$  and  $\vec{C}_y$ . Thus, the semantic similarity between tag  $x$  and tag  $y$  can be represented as the cosine similarity of  $\vec{C}_x$  and  $\vec{C}_y$ , namely  $\cos(\vec{C}_x, \vec{C}_y)$ . To establish the links between user tags and item tags, i.e.,  $E_0$  in the bi-partite graph  $G_0$ , we can set a threshold  $\lambda$ . Then, the edge between  $x$  and  $y$  is created if  $\cos(\vec{C}_x, \vec{C}_y) \geq \lambda$ . As well,

$\cos(\vec{c}_x, \vec{c}_y)$  is set as the element of connectivity matrix  $\mathbf{X}_0$  (refer to Figure 3). Accordingly,  $\lambda$  decides the density of  $\mathbf{X}_0$ . In our experiments, we will display how  $\lambda$  affects on the performance of our recommendation algorithm.

### 3. EXPERIMENTS

In this section, we evaluate the performance of our proposed graph-based algorithm through two applications of recommendation across heterogeneous domains.

#### 3.1 Experiment Setup

We first present some experiment settings including data sets, baseline methods and performance metrics.

##### 3.1.1 Data sets

To demonstrate the performance of our algorithm, we focus on the following two different types of data sets in our evaluation experiments.

##### 1. Weibo-Douban Data Set.

The first data set is collected from two Chinese websites: Weibo and Douban. Weibo is the largest Chinese Twitter which has more than 0.6 billion accounts and 90 million active users per day. Recommending various products to Weibo users can create plenty of business opportunities. On the other hand, Douban is a famous Chinese website of movie reviews. Although both Weibo users and Douban movies are profiled by a set of tags, the tags in these two websites are quite different (recall to Figure 1 in Section 1). Thus, recommending Douban movies to Weibo users is a typical recommendation problem across heterogeneous domains. To construct the bi-partite graphs for this problem, we define the edge weight between an object node and a feature node (the weight of  $E_i$ ,  $i=1,2$ ) by a *tag score*, which quantifies the extent to which a tag can characterize the user/movie. For a Weibo user, the tag score is computed by a local tag propagation algorithm proposed in [33]. For a Douban movie, the score of a tag is its tagging frequency for this movie which can be fetched directly from Douban website. From Douban website, we totally collected 4,028 movies whose average rating scores are above 7.5 (10 is the best) and review numbers are greater than 2000. Since most Weibo tags and Douban tags are both written in Chinese, we used Baike encyclopedia to explore the semantic similarity of these tags.

##### 2. Diabetes Data Set.

The second data set is collected from two diabetes social network websites, i.e., Diabetes1 and Diabetes2 introduced in Section 1. It can be obtained from our Lab’s website (<http://gdm.fudan.edu.cn/GDMWiki/Wiki.jsp?page=Network%20DataSet>). The former is dedicated to diabetes patients of Type I, Type II, and pre-diabetes, whereas the latter focuses on female diabetes patients, especially those with gestational diabetes. Our goal is to recommend discussion threads from Diabetes2 to the users of Diabetes1. Due to the different vocabulary used by the different websites, we can also regard this problem as recommendation across heterogeneous domains. In this data set, we randomly collected 5,000 users from Diabetes1. Each user is profiled by a set of keywords extracted from his/her historical posts on this website. From Diabetes2, we collected 2,790 posts (threads) each of which is also characterized by a set of keywords. Furthermore, to fully demonstrate the effectiveness of our proposed approach, we removed some tags shared by the two forums as a pre-processing step. When we constructed the bi-partite graphs, we set the edge

weights of  $E_i$  ( $i=1,2$ ) as the TF-IDF scores of the keywords for each user/post.

#### 3.1.2 The Ground Truth of Recommendation

Then, we introduce how to capture the ground truths of the two cross-domain recommendation tasks in our experiments.

##### 1. Douban Movie Recommendation.

Unlike some famous movie recommendation data set such as *MovieLens* (<https://movielens.org/>), there are no explicit rating/review data of Weibo users on Douban movies, it is difficult to directly obtain the ground truth of Weibo users’ preferences on Douban movies. Therefore, we have to rely on human assessments for generating ground truth of Douban movie recommendation. Specifically, we randomly selected 100 Weibo users from our data set as volunteers, to whom we recommended some Douban movies generated by the tested approaches. Then we asked each volunteer whether to accept the recommended movies or not. Each volunteer can give one of three answers, i.e., *yes*, *no* and *unknown*. We only took the move of *yes* as a *hit* recommendation. We took their average acceptance rates as the performance measure of the tested approaches.

##### 2. Diabetes Post Recommendation.

To generate the ground truth of a diabetes patient’s preferred posts, we first generated a representative vector for each Diabetes1’s user and Diabetes2’s post. Given a Diabetes1’s user, we collected all his/her published posts as a corpus from which a set of keywords were extracted. Each element of the user vector is the TF-IDF score of one keyword. So does a Diabetes2’s post vector. Then, we ranked all posts according to the cosine similarity of the user vector and the post vectors. Finally, we assume that the user will only accept top 30% of the posts whose cosine similarity is greater than 0. These posts were considered as recommendation ground truth of the user. Since all Diabetes keywords are English words, we used Wikipedia’s concepts to compute the semantic similarity of keywords.

#### 3.1.3 Baselines

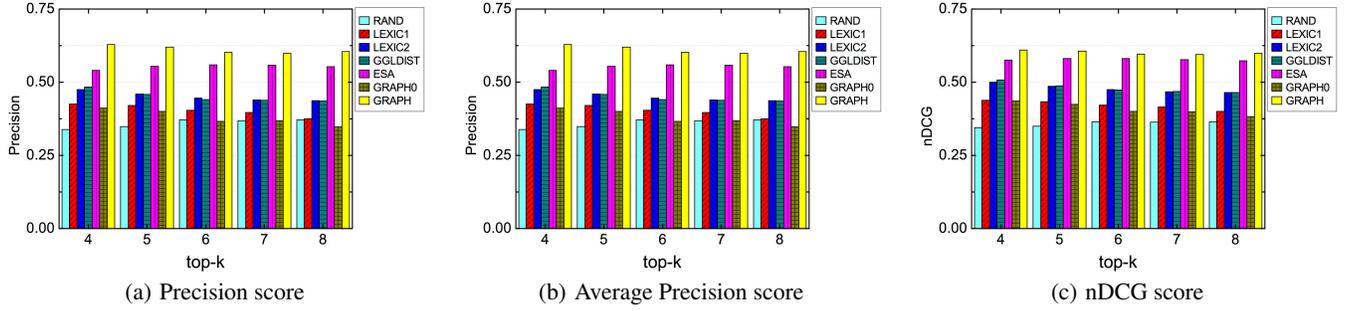
To emphasize the effectiveness of our approach, we also compared the performance of some state-of-the-art baselines.

##### 1. Random Recommendation.

It is the most naive method for recommendation. To let this baseline be more competitive, we randomly recommended items from some filtered candidates. In this naive baseline, for Douban movie recommendation, we randomly selected movies from those having more than 15,000 reviews and more than 8.8 rating scores. For Diabetes recommendation, we randomly recommended the posts from those having more than 2 keywords shared with the users. We denote this baseline as RAND.

##### 2. Lexical Matching.

It measures the similarity between a user and an item by computing the cosine similarity of their feature (tag/keyword) vectors. This recommendation scheme can be further divided into two baselines, i.e., LEXIC1 and LEXIC2. In LEXIC1, the vector’s element is set as 1 if the user/item has the corresponding feature, otherwise as 0. In LEXIC2, each element of the vector equals to the corresponding feature’s score to the user/item. Obviously, an item will not be recommended to a user by this method if they do not share any common feature.



**Figure 5: Human assessment performance of Douban movie recommendation. Each column represents a competitor method. The X-axis represents the number of recommended movies. It shows that our method (GRAPH) beats all baselines in each top-k cases.**

### 3. ESA Matching.

To highlight the effectiveness of our graph propagation approach, we adopted ESA scheme [11] as a baseline. Specifically, a user or an item are characterized by a profile vector that is the sum of concept vectors of user/item’s features. Then, we measure the similarity between a user and an item by the cosine similarity of their profile vectors.

### 4. Google Distance.

Besides ESA, Normalized Google Distance (NGD for short) [9] is also an effective measure on two terms’ semantic relatedness. For recommendation, the similarity between a user and an item can be represented by the NGD of their feature (tag/keyword) sets. Specifically, given a user  $u$  and an item  $v$ , suppose  $T_u$  is the in-neighbor concept set of  $u$ ’s features in the encyclopedia hyper-link graph, so is  $T_v$  for  $v$ ’s features. Then the NGD similarity of  $u$  and  $v$  is computed as

$$GS(u, v) = 1 - \frac{\max\{\log|T_u|, \log|T_v|\} - \log|T_u \cap T_v|}{\log N - \min\{\log|T_u|, \log|T_v|\}} \quad (4)$$

where  $N$  is the total number of concepts in the encyclopedia. We denote this baseline as GGLDIST.

#### 3.1.4 Performance Metrics

In our experiments, we use the following three metrics to quantify the acceptance rate of the users (Weibo users or diabetes users) to the top- $k$  recommended items (Douban movies or diabetes posts). The results of these metrics directly reflect the performance of all recommendation methods.

**Precision at rank  $k$  ( $Pr$ ).** It is defined as the proportion of top- $k$  recommended items that are accepted by the users.

**Average Precision at rank  $k$  ( $AP$ ).** It is the average precision score of top- $k$  recommended items and is defined as:

$$AP = \frac{\sum_{i=1}^k (P(i) \times rel(i))}{M} \quad (5)$$

where  $rel(i)$  is an indicator function. It equals to 1 if the  $i$ -th recommended item is accepted, otherwise 0.  $P(i)$  is the accepted proportion of top- $i$  items and  $M$  is the total number of accepted items in all top- $k$  items. Compared with  $Pr$ ,  $AP$  is more accurate to measure ranking performance.

**Normalized Discounted Cumulative Gain ( $nDCG$ ).** It is another popular metric to measure relevance level of search results

to the query in IR systems [17]. For a set of top- $k$  recommended items, the  $nDCG$  score can be calculated as:

$$nDCG = \frac{1}{Z} \sum_{i=1}^k \frac{2^{rel(i)} - 1}{\log_2(i + 1)} \quad (6)$$

where  $rel(i)$  is same as Eq. (5) and  $Z$  is a normalized factor. Compared with  $AP$ ,  $nDCG$  is more sensitive to rank position of recommended items. In general, a user pays less attention to the items listed behind, hence  $nDCG$  is better to evaluate recommendation performance.

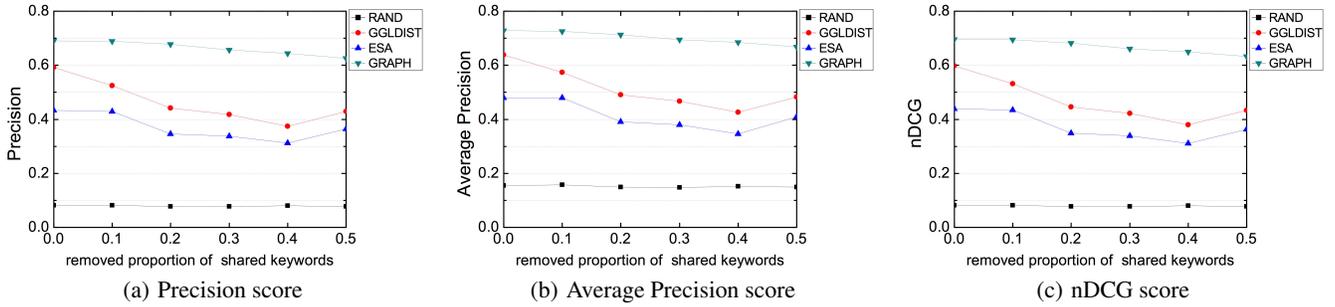
## 3.2 Evaluation Results

Next, we present our experimental results. In particular, we use Precision, Average Precision and  $nDCG$  as evaluation metrics to quantify the acceptance rate of the users to the top- $k$  recommended items, i.e., movies and posts. Note that we set the  $\alpha$  in Equation (2) to 0.99 when running our algorithm in the experiments. Small variations of  $\alpha$  will not affect the comparative performance of our approach versus the competitors. We also study the impact of the parameter  $\lambda$  on the overall performance, which is used to construct the connectivity matrix based on semantic matching.

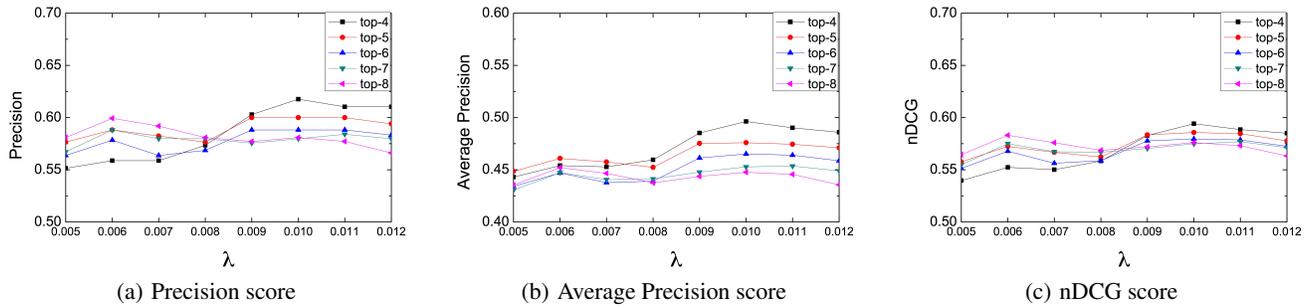
### 3.2.1 Douban Movie Recommendation

For Douban movie recommendation task, as mentioned before, the performance results were evaluated by surveying the 100 volunteers who have Weibo accounts in our data set. In order to highlight our graph-based approach’s superiority, we compared our approach with all baselines for this recommendation task. Furthermore, we selected an optimal value of  $\lambda$  ( $=0.01$ ) in our algorithm. The parameter study of  $\lambda$  will be introduced in the following subsection.

Figure 5(a)~(c) display the average scores of all competitors’ performance under the three metrics including the performance of recommending top-4 to top-8 movies. We only list the results from top-4 to top-8 due to the space limitation of the figures. The results of other top- $k$  recommendations also highlight our method’s superiority. In the figures, our graph-based method is denoted as GRAPH. GRAPH0 is a variation of our method where the weights of  $E_i$  ( $i = 0, 1, 2$ ) are set to 1 or 0 instead of concrete feature score (for  $E_1$  and  $E_2$ ) or semantic similarity value of two terms (for  $E_0$ ). From the results we find that all methods perform best when recommending top-4 movies. Naturally, good recommendation methods should rank the *hit* items to priority positions. Compared with other competitors, GRAPH almost always performs the best. The superiority of GRAPH over GRAPH0 justifies the advantage of fine-grained edge weight in the multi-partite graph. The superiority of ESA and GRAPH over LEXIC1/2 justifies the effec-



**Figure 6: The performance of all competitors on recommending top-4 diabetes posts. It shows that our method (GRAPH) outperforms the baselines no matter how many shared keywords across domains are removed.**



**Figure 7: Human assessment performance of graph-based method in different  $\lambda$ s. Each line depicts a top-k Douban movie recommendation, showing that 0.01 is the best threshold to decide the sparsity of  $E_0$  in the bi-partite graph.**

tiveness of concept-based semantic interpretation w.r.t. uncovering the latent relationships between users and items. The superiority of GRAPH over LEXIC1/2, GGLDIST and ESA indicates that, our graph-based method can discover more similarities between users and items that are originally neglected due to the gap of heterogeneous domains. The outperformance of our approach over these baselines should be mainly attributed to graph propagation principle.

### 3.2.2 Diabetes Post Recommendation

By statistics, we found that there are many shared keywords across the two diabetes domains. Specifically, the Jaccard coefficient of the keyword sets of two domains is 0.221. As a result, we first removed a fraction of shared keywords before running the algorithms in order to simulate the setting of cross-heterogenous-domain recommendation, i.e., few common keywords can be found in the two domains. Then, we tested the algorithms under different situations when a specific proportion of shared keywords are deleted. As introduced in Subsec. 3.1.2, the ground truth of diabetes post recommendation were generated according to the principle of LEXIC2. Consequently, we only compared GRAPH with RAND, GGLDIST and ESA for this recommendation task. Referring to the results of Douban movie recommendation, we only list the performance of top-4 diabetes posts since all approaches perform best when recommending top-4 items. The other top-k results will not affect the comparative performance of our proposed approach versus the competitors.

Figure 6(a)~(c) depict performance scores of the competitors as the function of removed proportion of shared keywords where the proportion varies from 0% to 50%. From the figures we can find that, although the performance of all methods except for RAND de-

cays evidently as more shared keywords are removed, GRAPH only decays a bit. Even when 50% of shared keywords are removed, our method’s performance can still stay above 0.6. It indicates that the graph-based method is more qualified for recommendation across heterogenous domains with rarely shared features.

### 3.2.3 Parameter Study

As mentioned before, the parameter  $\lambda$  in our algorithm decides the sparsity of  $E_0$  in the bi-partite graph  $G_0$ . That is, for two feature nodes belonging to two different domains respectively, an edge linking them is established only when the semantic similarity of the two features is bigger than  $\lambda$ . Then, how does it affect the performance of our graph-based algorithm? To answer this question, we tested the performance of our approach when the  $\lambda$  is set to different values.

Figure 7(a)~(c) show our approach’s performance as the function of  $\lambda$  on Douban movie recommendation. In the figures, each line corresponds to the performance of a top-k recommendation where k varies from 4 to 8. From the figures we can see the performance of our proposed approach is robust with respect to small variations in the value of  $\lambda$ . In particular, our approach gains the best performance in all metrics when  $\lambda = 0.01$ . The results of other k’s values also agree this point. On one hand, it is obvious that too sparse edges in  $G_0$  (larger  $\lambda$ ) are not beneficial to the recommendation performance because fewer connections across two domains can be derived through them. On the other hand, our experiment results indicate that more edges in  $G_0$  (smaller  $\lambda$ ) between user features and item features will not improve the recommendation performance either because potential noises will be imported by the additional edges with lower weights in  $G_0$ .

## 4. RELATED WORK

In this section, we survey the research works related to our work through the following three categories.

### 4.1 Social and Cross-domain Recommendation

By now, *Social Recommender Systems* have become the main part of recommender systems including content-based and structure-based mechanisms. Hannon et al. [13] proposed content-based user profiling based on CF for recommending new friends in Twitter. In [1], users are first represented by a frequency vector of hashtags and entities in tweets and then are recommended to with the URLs having similar profile vectors. In [3], consumers are recommended to based on the analysis of product reviews. For structure-based mechanism, the authors in [4, 8] emphasized that the structure of social links is an important clue to recommend new users. The social clues are effective on mitigating cold start problem in CF.

More recently, *cross-domain recommendation* has been a hot spot in recommendation field. Ignacio et al. [10] made a survey of emerged solutions for cross-domain recommendation and emphasized two major tasks. One is to exploit knowledge about the users and the items in the source domain for improving recommendation quality for the items in the target domain. The second task is to make joint recommendations for the items belonging to different domains. Many previous works on cross-domain recommendation focus on improving CF-based scheme. For example, the authors in [19, 32] tried to migrate the rating data from a dense auxiliary domain to alleviate the cold start problem in a sparse target domain resulting in the improvement of recommendation performance in the target domain. Besides, [2, 29] merged user profiles distributed in different domains for better recommendation. The settings in above works are different from our problem setting. The cross-domain setting of Zhang et al.'s work [34] is similar to ours. But they correlated Facebook users to eBay products through Facebook's page categories and eBay taxonomy other than the semantic similarities between features in the heterogenous domains.

### 4.2 Transfer Learning across Heterogenous Domains

Transfer learning has been extensively studied in the past decade. It aims to improve a learning task in a target domain by using the knowledge transferred from other domain in which a related task is known [23]. Recently, transfer learning techniques have been widely applied to mitigate the sparsity problem of collaborative filtering in cross-domain recommendation.

In [19], Li et al. proposed a transfer learning approach that performs a co-clustering strategy on the rating matrix of an auxiliary domain with high rating density, and discovers rating patterns at the cluster level. Assuming that user rating behavior is similar in two domains, the approach establishes relations between domains based on the found rating patterns. In their other work [20], Li et al. extended the previous approach by means of a probabilistic model in which a user or item belonging to a particular cluster is not binary, but is described in terms of probability density function. In this case a common model is built from the ratings of all the considered domains, without requiring a dense source domain. Also addressing the sparsity problem in CF-based recommendation, W. Pan et al. [24] transferred the rating knowledge from some auxiliary data source in binary form to a target numerical rating matrix through a novel framework of transfer by collective factorization. In all above works, the user-item relations, i.e., users' ratings on items exist in both domains which is different to our scenario. Y. Zhu et al. [37] also applied transfer learning method to learn image classification

by using the knowledge of document/image labels in auxiliary domains. In this work, relations between documents and images are captured by their co-occur tags. B. Cao et al. [7] proposed a non-parametric Bayesian framework for solving the collective link prediction, which allows knowledge to be adaptively transferred across heterogeneous tasks while taking into account the similarities between tasks. These methods are not suitable to be applied in our setting, where we aim to build the connections between heterogeneous entities across different domains instead of learning multiple models.

### 4.3 Semantic Relatedness Measurement

In NLP, measuring semantic relatedness of two terms or entities is an important task. Many prior works utilized the lexical concepts in WordNet's taxonomy [18] based on the deepest point in the taxonomy [31] or information content [26]. To expand concept coverage, many researchers took Wikipedia as the knowledge base of semantic interpretation. M. Strube et al. [28] and D. Milne et al. [22] used the taxonomy and the Normalized Google Distance [9] in Wikipedia to compute semantic relatedness, respectively. E. Gabrilovich et al. [11] proposed a widely applied model of semantic interpretation, i.e., Explicit Semantic Analysis which is also based on the relations between concepts in online encyclopedia. More recently, T. Mikolov et al. [30] proposed *word embedding* which learns a distributed representation for a word through neural network. Then the semantic relatedness of two words can be computed by the distance of two word vectors. This approach is effective but costly in computation and depends training corpus.

## 5. CONCLUSION

In this paper, we address a new challenging problem of recommendation across heterogenous domains. That is, recommending the items in one domain to the users in another domain when the features of the two domains rarely overlap, as well as no user-item interactive relations in each domain can be obtained. To this end, we first capture the semantic relations between user features and item features (tags/keywords) through ESA-based concept interpretation from online encyclopedias. Then, we propose a novel graph-based algorithm to discover the hidden similarity between two entities (users/items) via propagation principle. We not only exploit our algorithm with solid proofs, but also justify our approach's performance superiority over the state-of-the-art competitors through extensive evaluations. The results demonstrate our approach's merits to many real world applications of cross-domain recommendation.

## 6. ACKNOWLEDGEMENTS

This paper was supported by the National NSFC (No.61472085, 61171132, 61033010), by National Key Basic Research Program of China under No.2015CB358800, by Basic research project of Shanghai science and technology innovation action plan under No.15JC1400900, and by Shanghai Science and Technology Development Funds (13dz2260200, 13511504300). This paper was also partially supported by an IBM Faculty Award.

## 7. REFERENCES

- [1] F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In *Proc. of WebSci*, 2011.

- [2] F. Abel, E. Herder, G.-J. Houben, N. Henze, and D. Krause. Cross-system user modeling and personalization on the social web. In *Proc. of UMUI*, 2013.
- [3] S. Aciar, D. Zhang, S. Simoff, and J. Debenham. Recommender system based on consumer product reviews. In *Proc. of WI*, 2006.
- [4] M. J. Brzozowski and D. M. Romero. Who should i follow? recommending people in directed social networks. In *Proc. of CSCW*, 2011.
- [5] R. Burke. Hybrid web recommender systems. *LNCS*, 4321:377–408, 2007.
- [6] I. Cantador, A. Bellogin, and D. Vallet. Content-based recommendation in social tagging systems. In *Proc. of Recommender System*, 2010.
- [7] B. Cao, N. N. Liu, and Q. Yang. Transfer learning for collective link prediction in multiple heterogenous domains. In *Proc. of ICML*, 2010.
- [8] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make new friends but keep the old, recommending people on social networking sites. In *Proc. of CHI*, 2009.
- [9] R. L. Cilibrasi and P. M. Vitanyi. The google similarity distance. *Knowledge and Data Engineering, IEEE Transactions on*, 19(3):370–383, 2007.
- [10] I. Fernandez-Tobias, I. Cantador, M. Kaminskas, and F. Ricci. Cross-domain recommender systems: A survey of the state of the art. 2011.
- [11] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of IJCAI*, 2007.
- [12] L. Gall and François. Powers of tensors and fast matrix multiplication. In *Proc. of ISSAC*, 2014.
- [13] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proc. of RecSys*, 2010.
- [14] Hazewinkel and Michiel. *Taylor series*. Encyclopedia of Mathematics, Springer, 2001.
- [15] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proc. of SIGIR*, 2003.
- [16] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 2:89 - 115, 2004.
- [17] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20:422-446, 2002.
- [18] F. L., G. E., and Matias. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA., 1998.
- [19] B. Li, Q. Yang, and X. Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *Proc. of IJCAI*, 2009.
- [20] B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proc. of ICML*, 2009.
- [21] Z. Lu, E. Zhong, L. Zhao, E. Xiang, W. Pan, and Q. Yang. Selective transfer learning for cross domain recommendation. In *Proc. of SDM*, 2013.
- [22] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proc. of AAAI*, 2008.
- [23] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE TKDE*, 22(10):1345 - 1359, 2010.
- [24] W. Pan, N. N. Liu, E. W. Xiang, and Q. Yang. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *Proc. of IJCAI*, 2011.
- [25] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007.
- [26] P. Resnick. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of IJCAI*, 1995.
- [27] S. Sen, J. Vig, and J. Riedl. Tagommenders: Connecting users to items through tags. In *Proc. of WWW*, 2009.
- [28] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proc. of AAAI*, 2006.
- [29] M. Szomszor, H. Alani, I. Cantador, K. OHara, and N. Shadbolt. Semantic modelling of user interests based on cross-folksonomy analysis. In *Proc. of ISWC*, 2008.
- [30] M. Tomas, S. Ilya, C. Kai, C. Greg, and D. Jeffrey. Distributed representations of words and phrases and their compositionality. In *arXiv:1310.4546*, 2013.
- [31] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proc. of ACL*, 1994.
- [32] W. C. Wynne, H. Mong, and L. Lee. Making recommendations from multiple domains. In *Proc. of SIGKDD*, 2013.
- [33] D. Yang, Y. Xiao, H. Tong, J. Zhang, and W. Wang. An integrated tag recommendation algorithm towards weibo user profiling. In *Proc. of DASFAA*, 2015.
- [34] Y. Zhang and M. Pennacchiotti. Predicting purchase behaviors from social media. In *Proc. of WWW*, 2013.
- [35] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Scholkopf. Ranking on data manifolds. In *Proc. of NIPS*, 2003.
- [36] T. C. Zhou, H. Ma, M. R. Lyu, and I. King. Userrec: A user recommendation framework in social tagging systems. In *Proc. of AAAI*, 2010.
- [37] Y. Zhu, Y. Chen, Z. Lu, S. J. Pan, G.-R. Xue, Y. Yu, and Q. Yang. Heterogeneous transfer learning for image classification. In *Proc. of AAAI*, 2011.