Semantic Social Network Analysis by Cross-Domain Tensor Factorization

Makoto Nakatsuji, Qingpeng Zhang, *Member, IEEE,* Xiaohui Lu, Bassem Makni, and James A. Hendler, *Fellow, IEEE*

***Abstract*— Analyzing "what topics" a user discusses with others is important in social network analysis. Since social relationships can be represented as multiobject relationships (e.g., those composed of a user, another user, and the topic of communication), they can be naturally represented as a tensor. By factorizing the tensor, we can perform *communication prediction* that predicts links among users and the topics discussed among them. The prediction accuracy, however, is often inadequate for applications because: 1) users usually discuss a variety of topics, and thus the prediction results tend to be biased toward popular domains and 2) topics that are rarely discussed among users trigger the sparsity problem in tensor factorization. Our solution, cross-domain tensor factorization (CrTF), first determines the topic domain by analyzing communication logs among users using the DBpedia knowledge base and creates a tensor composed of users, other users, and the topics of communication for each domain; it avoids strong bias toward particular domains. It then simultaneously factorizes tensors across domains while integrating semantics from DBpedia into factorizations; this solves the sparsity problem. Experiments using Twitter data sets show that CrTF achieves higher accuracy than the state-of-the-art tensor-based methods and extracts key topics and social influencers for each domain.**

***Index Terms*— Computing/semantic web, mathematics/ prediction algorithms, professional communication/recommender systems, professional communication/social network services.**

I. Introduction

SOCIAL network analysis [21], [27] has become an important business area, because social aspects can enhance Web applications and can be used by marketing experts in developing their strategies. For example, recent recommendation studies indicated that users' future preferences are influenced by the topics of discussion with their friends [9], [13]. In fact, Amazon and Facebook recently have joined forces to show users their Facebook friends on Amazon and enable users to find out what their friends like in terms of movies, music, and so on. In addition, social networks have been used in political campaigns [26]. Political parties want to know people's opinions about their policies and to identify topics discussed by social network users and social media influencers. For example, Obama and Romney campaign staff regularly engaged in "Twitter duels" online, with reporters and activists being the intended audience.[1] Therefore, analyzing what topics users will be interested in through communication with other users is important for political parties to enhance their policies.

One approach to doing such analyses is to represent social relationships on the Web as multiobject relationships. Such social relationships can be seen in social networks formed in Twitter composed of users who retweet tweets, topics in the tweets, and users who wrote the original tweets (retweeted users). Here, a topic represents a subject matter of a tweet and is formed by a phrase; we note that the term "topic" in this paper is not identical to the "topic" used in the research studies in topic modeling [4]. Phrases representing topics can be estimated from tweets by using the semantic Web study that accurately extracts tweets that mention a targeted domain [15] or DBpedia spotlight, which is a tool for automatically annotating mentions of DBpedia resources in text [14]. Tensors are suitable representations for such multiobject relationships. The previous 2-D (user-retweeted user) matrix can be turned into a 3-D (user-topic-retweeted user) tensor. The factorization of this tensor leads to a compact model of the data; it clusters users with the topics discussed among them. This enables *communication prediction* that predicts social relationships with the topics discussed among users. Previous 2-D social network analyses by matrix factorization or random walk methods [6], [8] cannot perform communication prediction with the topics discussed.

The predictions of current tensor factorization schemes, however, can fail for two reasons. First, users discuss many different topics in different domains, so the prediction results often tend to be biased toward the domains that are the most common [16]. Here, a domain is defined by a set of topics that are related to a key topic discussed among a social network. For example, consider two social networks related to key topics "Rahul Gandhi" and "Narendra Modi," the political leaders of the Indian National Congress (INC) and Bharatiya Janata Party (BJP), the two major political parties in India. In the discussions among users on these two social networks, one could find several common topics; however, many other topics would be strongly biased toward one of those social networks. We can merge social relationships in those two social networks to create a merged social network and factorize a tensor created for the merged social network to compute the communication prediction. The results, however, would tend to be biased toward one of those domains that has many more observations than the other. As a result, the overall

Manuscript received June 19, 2017; accepted July 20, 2017. Date of publication September 1, 2017; date of current version November 21, 2017. *(Corresponding author: Makoto Nakatsuji.)*

M. Nakatsuji is with NTT Resonant Inc., Tokyo 108-0023, Japan (e-mail: nakatuji@nttr.co.jp).

Q. Zhang is with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong, and also with the Shenzhen Research Institute, City University of Hong Kong, Shenzhen 518057, China.

X. Lu, B. Makni, and J. A. Hendler are with the Tetherless World Constellation, Troy, NY 12180 USA.

Digital Object Identifier 10.1109/TCSS.2017.2732685

[1] http://www.epolitics.com/2014/04/09/

2329-924X © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

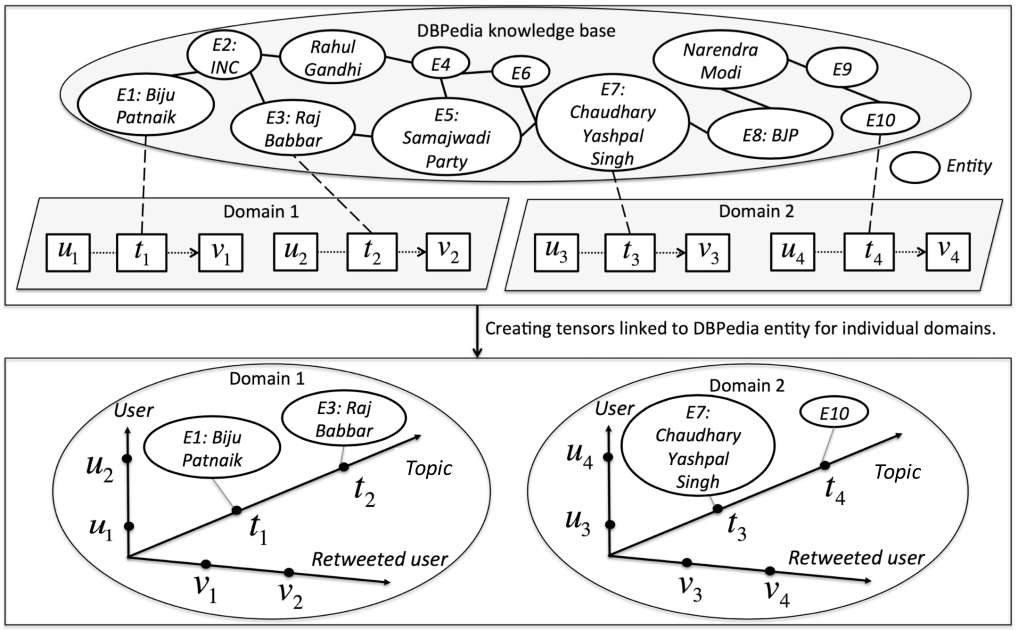


Fig. 1. Creating tensors for individual topic domains.

prediction accuracy falls dramatically. The other problem is that there are many sparse topics that are rarely observed in each social network. According to the long-tail theory [1], a few topics are discussed often in a social network but most topics tend to be rarely discussed. As a result, there are only a few observed elements indicative of possible multiobject relationships in a tensor. This leads to the sparsity problem in tensor factorization [20].

To solve the above-mentioned problems, we propose cross-domain tensor factorization (CrTF); it performs a coupled analysis of tensors created from two different social networks as well as incorporates semantic knowledge extracted from DBpedia [3] into tensor factorization. It is found based on two ideas.

1) It extracts a social network created for a targeted topic domain and creates a tensor for each social network. The procedure is as follows: it first crawls the tweets that include the name of "the representative entity" in the DBpedia knowledge base (e.g., "Rahul Gandhi" if we want to analyze social networks for that politician). The key topic in extracting a social network should be selected from the DBpedia entity and we call this entity as "the representative entity" for this social network. It next extracts the entities that are within a few hops from the representative entity in the DBpedia knowledge base. It then extracts topics in tweets that match the name of the above-mentioned entities and links those topics to DBpedia entities. Finally, it creates tensors for the social networks of the two domains linked with DBpedia entities. By using separate tensors for each domain, it can avoid strong biases toward particular domains.

   Fig. 1 shows an example of social relationships among three objects: a user who retweets tweets, topics in the tweets, and retweeted users who wrote the original tweets. For example, user u_1 retweeted a tweet containing topic t_1 described by retweeted user v_1. Here, we can restrict the vocabulary of the topics to the names of the entities that are within two hops from the representative entity "Rahul Gandhi," in this case, $E1$, $E2$, $E3$, $E4$, $E5$, and $E6$. As a result, CrTF can create a tensor linked to semantic entities; it focuses on the social network for "Rahul Gandhi."
2) It uses semantic knowledge behind the sparse topics in tensor factorization to solve the sparsity problem. It first propagates sparse topics to neighboring entities of the entity linked by that topic in the DBpedia entity space. It next creates a semantically augmented tensor that adds the relationships composed of those neighboring entities propagated from sparse topics to the tensor created in the above-described idea 1 for each domain. It then simultaneously factorizes the individual tensors and semantically augments tensors in two domains. During factorization, it incorporates the semantic biases generated from the propagated entities in the augmented tensors into the features for the sparse topics in the individual tensors. This approach solves the sparsity problem in tensor factorization across domains.

   In Fig. 1, suppose that there are only a few observations for topics t_1, t_2, and t_3. In such a case, the prediction accuracy may fail, because the factorized results tend not to reflect such sparse topics. CrTF propagates observations for sparse topics t_1 and t_2 to the neighboring entity $E2$ of entities $E1$ and $E3$ that are linked by t_1 and t_2. CrTF applies biases from entity $E2$ to t_1 and t_2 when the individual tensor for the "Rahul Gandhi" domain is factorized. It also applies biases to t_2 and t_3 by using entity $E5$, which neighbors $E3$ and $E7$,

when the individual tensor for "Rahul Gandhi" and that for "Narendra Modi" are factorized. In this way, CrTF analyzes sparse topics over the semantic entity space across domains and solves the sparsity problem.

We applied our ideas to variational nonnegative tensor factorization (VNTF), an extension of variational nonnegative matrix factorization (VNMF) [5]. We used VNTF because it is based on Poisson–Gamma priors for parameters and is suitable for predicting communication frequencies among users that often follow long-tail distributions.

We evaluated CrTF by using: 1) retweeting relationships among users focused on political parties in India and 2) those in the Middle East.[2] To the best of our knowledge, this is the first study that makes communication predictions with topics discussed among users and uses a semantic space in factorizing the tensors created across domains to improve prediction accuracy. The results show that CrTF outperforms the state-of-the-art methods, including VNTF, generalized coupled tensor factorization (GCTF) [7], and semantic data representation for tensor factorization (SRTF) [17]. It can also cluster key topics and social media influencers with semantic labels for each domain.

This paper is organized as follows. Section II describes related work and Section III introduces the background of this paper. Section IV explains our method in detail and Section V evaluates our method. Finally, Section VI concludes this paper.

II. Related Work

Tensor factorization methods are one of core AI technologies like deep neural networks [24] and have recently been used in various applications such as social network analysis [12] and recommendation [22]. Among the recent proposals, semantic data representation for tensor factorization (SRTF) [17], and its extension, semantic-sensitive tensor factorization (SSTF) [18] incorporates semantic knowledge in terms of taxonomies/vocabularies extracted from linked open data (LOD) in tensor factorization. It can solve the sparsity problem by providing semantic biases to the feature vectors for sparse objects in multiobject relationships. However, SRTF and SSTF are not designed for cross-domain analysis even though LOD can be used for mediating distributed objects in different service domains [3]. GCTF [7], [25] and a few other studies [23], [29] have tried to incorporate extra information into tensor factorization by simultaneously factorizing observed tensors (e.g., user-item-time tensor) and matrices (e.g., user-tag matrix) representing extra information via shared objects (users). GCTF, however, has no ability to extract social networks for individual topic domains. It can be applied to our idea; factorizing the tensors created from individual social networks, while integrating tensors via shared objects (users). There are, however, no tensor methods that exploit semantics to solve the sparsity problem across domains.

More recently, we proposed semantic-sensitive simultaneous tensor factorization S^3TF [19] that includes the semantics behind objects into tensor factorization, and thus analyzes users' activities across different services. S^3TF, however, uses Gaussian priors for its parameters, and thus is not suitable for predicting communication frequencies among users that often follow long-tail distributions.

[2]The data sets and our MATLAB code can be acquired by mailing the authors.

III. Preliminaries

Here, we explain VNMF [5], since we base our ideas on its framework.

First, let us introduce the notation used in this paper and used to explain VNMF. VNMF can deal with the biobject relationships formed by user u_m and retweeted user v_n; u_m denotes the mth user who retweets tweets that were originally written by user v_n, which is the nth retweeted user. Matrix factorization assigns a D-D latent feature vector to each user and retweeted user, denoted as $\mathbf{u}_m$ and $\mathbf{v}_n$, respectively. Here, $\mathbf{u}_m$ and $\mathbf{v}_n$ are D-D column vectors. We define a matrix $\mathbf{R}$, whose element $r_{m,n}$ represents the frequencies of communications made by the biobject relationships formed by user u_m and retweeted user v_n. We can approximate $\mathbf{R}$ as the inner product of the above two vectors as follows:

$$r_{m,n} \approx \langle \mathbf{u}_m, \mathbf{v}_n \rangle \equiv \sum_{i=1}^{D} u_{i,m} \cdot v_{i,n}. \tag{1}$$

Index i represents each vector's ith element. The matrix representations of $\mathbf{u}_m$ and $\mathbf{v}_n$ are $\mathbf{U} \equiv [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]$ and $\mathbf{V} \equiv [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N]$, respectively.

Now, let us explain VNMF that computes the unobserved elements in $\mathbf{R}$ on the basis of variational Bayes [2]. VNMF computes the unobserved frequencies $\mathbf{R}$ by marginalizing over model parameters $\mathbf{U}$, $\mathbf{V}$ and hyperparameters $\Theta \equiv (A_u, A_v, B_u, B_v)$

$$p(\mathbf{R}|\Theta) = \int d\mathbf{U} d\mathbf{V}\, p(\mathbf{R}|\mathbf{U}, \mathbf{V})\, p(\mathbf{U}, \mathbf{V}|\Theta) \tag{2}$$

where A_u, A_v, B_u, and B_v are hyperparameters used to generate gamma distributions as explained below. VNMF computes the approximating marginal log-likelihood $\log p(\mathbf{R}|\Theta)$ based on the variational Bayes method. We here summarize its procedure as follows (please see [5] for the detailed procedure and we define .âˆ— and ./ as elementwise matrix multiplication and division, respectively).

1) Initialize feature matrices for users $\mathbf{U}$ and retweeted users $\mathbf{V}$ as well as logarithmic feature matrices for users $\mathbf{U}'$ and retweeted users $\mathbf{V}'$.
   For example, $\mathbf{U}$ and $\mathbf{U}'$ are prepared as $\mathbf{U}^{(0)} = \mathbf{U}'^{(0)} \sim g(A_u, B_u./A_u)$, where the function $g(A_u, B_u./A_u)$ generates a matrix whose elements have random numbers from a gamma distribution with a shape matrix A_u and scale matrix $B_u./A_u$. A_u and B_u are hyperparameters for $\mathbf{U}$. $\mathbf{V}$ ($\mathbf{V}'$) is initialized in the same way.
2) Repeat steps (a)–(c) L times until the approximating marginal log-likelihood is converged. Implementally, we set L to be the maximum iteration count.
   a) Compute Σ_u and Σ_v, which are sources of sufficient statistics for $\mathbf{U}$ and $\mathbf{V}$, respectively. For example, VNMF computes Σ_u as

$$\Sigma_u = \mathbf{U}'. * (((\mathbf{X}. * \mathbf{R})./(\mathbf{U}'\mathbf{V}'))\mathbf{V}'^T)$$

where $\mathbf{X}$ is a matrix whose element $x_{m,n}$ is 1 if $r_{m,n}$ is observed and 0 if not. Σ_v is computed similarly.

b) Update $\mathbf{U}$ and $\mathbf{V}$. For example, $\mathbf{U}$ is updated as

$$\mathbf{U} = \alpha_u . * \beta_u$$

where $\alpha_u = A_u + \Sigma_u$ and $\beta_u = 1./(A_u./B_u + \mathbf{R}. * \mathbf{V}^T)$. $\mathbf{V}$ is updated similarly.

c) Update $\mathbf{U}'$ and $\mathbf{V}'$. For example, $\mathbf{U}'$ is updated as $\mathbf{U}' = \exp(\Psi(\alpha_u)). * \beta_u$. Ψ is the digamma function [2]. $\mathbf{V}'$ is updated similarly.

3) Finally, VNMF computes the unobserved elements in $\mathbf{R}$ by applying $\mathbf{U}$ and $\mathbf{V}$ to (1).

IV. Method

We now explain our method in detail. First, we extend VNMF to VNTF, which is an instance of the CANDECOMP/PARAFAC decomposition [11]. Then, we explain our CrTF that applies the VNTF to perform a coupled analysis of tensors created from two different social networks as well as incorporates semantic knowledge extracted from DBpedia. Please see Table I also.

A. Variational Nonnegative Tensor Factorization

Here, we explain VNTF. We use the Variational Bayes, because it is based on Poisson–Gamma priors for parameters and is suitable for predicting communication frequencies among users that often follow long-tail distributions. As long as the communication frequency follows such a low power, the Poisson–Gamma priors are widely applicable for all the communication frequency prediction cases.

First, let us introduce the notation used in VNTF in addition to the notations used by VNMF. This paper deals with the multiobject relationships formed by user u_m, retweeted user v_n, and topic t_k; the mth user u_m retweets tweets that were originally written by the nth retweeted user v_n and that include the kth topic t_k. Tensor factorization assigns a D-D latent feature vector to each user, retweeted user, and topic, denoted as $\mathbf{u}_m$, $\mathbf{v}_n$, and $\mathbf{t}_k$, respectively. Here, $\mathbf{u}_m$, $\mathbf{v}_n$, and $\mathbf{t}_k$ are D-length column vectors, respectively. We define a tensor $\mathcal{R}$, whose element $r_{m,n,k}$ represents the frequencies made for the multiobject relationships formed by user u_m, retweeted user v_n, and retweeted topic t_k. We can approximate $\mathcal{R}$ as the inner product of the above-mentioned three vectors as follows:

$$r_{m,n,k} \approx \langle \mathbf{u}_m, \mathbf{v}_n, \mathbf{t}_k \rangle \equiv \sum_{i=1}^{D} u_{i,m} \cdot v_{i,n} \cdot t_{i,k}. \quad (3)$$

Index i represents each vector's ith element. The matrix representations of $\mathbf{u}_m$, $\mathbf{v}_n$, and $\mathbf{t}_k$ are $\mathbf{U} \equiv [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]$, $\mathbf{V} \equiv [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N]$, and $\mathbf{T} \equiv [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_K]$, respectively.

VNTF computes the unobserved frequencies $\mathbf{R}$ by marginalizing over model parameters $\mathbf{U}$, $\mathbf{V}$, $\mathbf{T}$ and hyperparameters $\Theta \equiv (A_u, A_v, A_t, B_u, B_v, B_t)$

$$p(\mathbf{R}|\Theta) = \int d\mathbf{U} d\mathbf{V} d\mathbf{T} p(\mathbf{R}|\mathbf{U}, \mathbf{V}, \mathbf{T}) p(\mathbf{U}, \mathbf{V}, \mathbf{T}|\Theta) \quad (4)$$

where A_u, A_v, A_t, B_u, B_v, and B_t are hyperparameters used to generate a gamma distribution as explained in the following.

TABLE I
Definition of Main Symbols

Symbols	Definitions
$\mathcal{R}$	Tensor that includes the frequencies of communications made by users to retweeted users with retweeted topics.
$r_{m,n,k}$	The frequencies of communications made by the multi-object relationships formed by user u_m, retweeted user v_n, and retweeted topic t_k.
$\mathbf{U}$	Feature matrices for users.
$\mathbf{V}$	Feature matrices for retweeted users.
$\mathbf{T}$	Feature matrices for retweeted topics.
D	Number of dimensional latent feature vectors.
Σ_u	Source of sufficient statistics for $\mathbf{U}$.
Σ_v	Source of sufficient statistics for $\mathbf{V}$.
Σ_t	Source of sufficient statistics for $\mathbf{T}$.
$\mathcal{X}$	A tensor whose element $x_{m,n,k}$ is 1 if $r_{m,n,k}$ is observed and 0 if not.
$\mathbb{E}$	A set vocabulary of DBpedia entities that are a few hops away from the representative entity for the targeted domain.
$\mathcal{R}^d$	A tensor composed of a user u_m who retweeted a tweet, the topic t_k^d of the tweet, and another user v_n^d who wrote the original tweet for the particular domain d.
$\mathcal{A}^d$	An augmented tensor for domain d.
$\mathbb{T}^d$	The set of sparse topics in domain d.
δ	A parameter used to determine the number of sparse topics in $\mathbb{T}^d$.
e_j	An entity linked to a topic t_j^d.
$f(e_j)$	The function that returns the set of neighboring entities of e_j in DBpedia network.
ne_s	The s-th neighboring entity in set $\bigcup_{t_j^d \in \mathbb{T}^d} f(e_j)$.
$\mathbf{u}_m$	m-th user feature vector.
$\mathbf{v}_n^d$	n-th retweeted user's feature vector in domain d.
$\mathbf{t}_k^d$	k-th retweeted topic's feature vector in domain d.
t_s^d	s-th sparse topic in domain d.
$\mathbf{c}_j^d$s	Feature vectors for neighboring entities of the entity linked by t_s^d in domain d.
$\mathbf{V}^d$	Feature matrices for retweeted users in domain d.
$\mathbf{T}^d$	Feature matrices for retweeted topics in domain d.
$\mathbf{C}^d$	Semantically biased feature matrices for retweeted topics in domain d.
Σ_v^d	Source of sufficient statistics for $\mathbf{V}^d$ in domain d.
Σ_t^d	Source of sufficient statistics for $\mathbf{T}^d$ in domain d.
Σ_c^d	Source of sufficient statistics for $\mathbf{C}^d$ in domain d.

Now, let us explain VNTF that computes the unobserved elements in $\mathcal{R}$ on the basis of variational Bayes [2] in the same way the VNMF does. It works as follows.

1) Initialize feature matrices for users $\mathbf{U}$, retweeted users $\mathbf{V}$, and retweeted topics $\mathbf{T}$ as well as logarithmic feature matrices for users $\mathbf{U}'$, retweeted users $\mathbf{V}'$, and retweeted topics $\mathbf{T}'$. For example, $\mathbf{U}$ is prepared as $\mathbf{U}^{(0)} = \mathbf{U}'^{(0)} \sim g(A_u, B_u./A_u)$, where the function $g(A_u, B_u./A_u)$ generates a matrix whose elements have random numbers from a gamma distribution with a shape matrix A_u and scale matrix $B_u./A_u$. A_u and B_u are hyperparameters for $\mathbf{U}$. $\mathbf{V}$ ($\mathbf{V}'$) and $\mathbf{T}$ ($\mathbf{T}'$) are initialized in the same way.
2) Repeat steps (a)–(c) L times, where L is the maximum iteration count.
   a) Compute Σ_u, Σ_v, and Σ_t, which are sources of sufficient statistics for $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{T}$, respectively. In computing Σ_u, VNTF divides a tensor into frontal matrices $\mathbf{R}_{:,n,:}$ (the user-topic adjacency matrix for retweeted user v_n) and $\mathbf{R}_{:,:,k}$ (the user-retweeted user adjacency matrix for topic t_k),

computes sources of sufficient statistics for those matrices in the same way as VNMF, and summarizes the results as follows:

$$\Sigma_u = \frac{1}{2}\mathbf{U}'.*\left(\left(\sum_n((\mathbf{X}\mathbf{R}_n)./(\mathbf{U}'\mathbf{T}'))\mathbf{T}'^T\right) + \left(\sum_k((\mathbf{X}\mathbf{R}_k)./(\mathbf{U}'\mathbf{V}'))\mathbf{V}'^T\right)\right) \quad (5)$$

where $\mathbf{X}\mathbf{R}_n \equiv \mathbf{X}_{:,n,:} .* \mathbf{R}_{:,n,:}$ and $\mathbf{X}\mathbf{R}_k \equiv \mathbf{X}_{:,:,k} .* \mathbf{R}_{:,:,k}$. $\mathcal{X}$ is a tensor whose element $x_{m,n,k}$ is 1 if $r_{m,n,k}$ is observed and 0 if not. Σ_u is computed from the two kinds of frontal matrices, so it is adjusted by dividing the numerator by 2. Σ_v (Σ_t) is computed similarly.

b) Update $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{T}$. For example, $\mathbf{U}$ is updated using frontal matrix $\mathbf{R}_{m,:,:}$ in the same way the VNMF did

$$\mathbf{U} = \alpha_u .* \beta_u$$

where $\alpha_u = A_u + \Sigma_u$ and $\beta_u = 1./\left(A_u./B_u + \frac{1}{2}\sum_m\left(\mathbf{R}_{m,:,:}\mathbf{V}^T + \mathbf{R}_{m,:,:}\mathbf{T}^T\right)\right)$. $\mathbf{U}$ is updated with the inner products of two different combinations of matrices, $\mathbf{R}_{m,:,:}$ and $\mathbf{V}$, and $\mathbf{R}_{m,:,:}$ and $\mathbf{T}$. Therefore, it is adjusted by dividing the numerator by 2. $\mathbf{V}$ ($\mathbf{T}$) is updated similarly.

c) Update $\mathbf{U}'$, $\mathbf{V}'$, and $\mathbf{T}'$ in the same way as VNMF (e.g., $\mathbf{U}'$ is updated as $\mathbf{U}' = \exp(\Psi(\alpha_u)) .* \beta_u$. Ψ is the digamma function [2]). $\mathbf{V}'$ ($\mathbf{T}'$) is updated similarly.

3) Finally, VNTF computes the unobserved elements in $\mathcal{R}$ by applying $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{T}$ to (3).

B. Cross-Domain Tensor Factorization

Now, we explain the CrTF in this section.

1) Creating a Tensor for Each Retweeting Network: Tweets are written in natural language and could be about anything. Hence, they could be on any topic as well. As such, CrTF links topics in tweets to DBpedia entities of our target domain while removing unrelated tweets. We used the results of a semantic Web study [15] that accurately extracts blog entries that mention a targeted domain as follows.

1) CrTF crawls retweets that include the name of the representative entity to determine the target topic domain. The representative entity should be chosen by experts who want to analyze the retweeting network for that domain. For example, they can set "Rahul Gandhi" as the representative entity, as explained using Fig. 1.
2) It crawls DBpedia entities that are a few hops away from the representative entity and creates a set vocabulary of entities $\mathbb{E}$ for the targeted domain. Here, we consider that one hop is from a subject entity to an object one and vice versa in a DBpedia triple. For example, if the subject is "Rahul Gandhi," the predicate is "Political party," and the object is "INC," it considers "INC" to be one hop from "Rahul Gandhi" and adds "INC" to $\mathbb{E}$. We also remove some Stop-words (words that are too ambiguous to indicate the domain) such as "Agent" and "Website" from $\mathbb{E}$.
3) It extracts the topics from tweets crawled in step 1. If the tweets include the name of E_i in $\mathbb{E}$, we consider that those tweets describe a topic related to E_i. This step disambiguates topics that may have several meanings, because the tweets include both the name of a representative entity and the names of entities that are semantically related to the representative one.

Then, it creates a tensor $\mathcal{R}^d$ composed of a user u_m who retweeted a tweet, the topic t_k^d of the tweet, and another user v_n^d who wrote the original tweet for the particular domain d. We assume that user u_m can be in several domains. This assumption is natural, because users tweet on a diverse range of topics. If u_m retweets nothing in domain d, the value for element $r_{m,n,k}^d$ is not assigned (the value is "N/A"). The number of users is M. The number of retweeted users and number of topics in domain d are N_d and K_d, respectively. CrTF also creates an original tensor, $\mathcal{R}$ that merges the tensors created for each domain. The number of users, number of retweeted users, and number of topics in $\mathcal{R}$ are M, $N = \sum_d N_d$, and $K = \sum_d K_d$, respectively.

2) Tensor Augmentation: CrTF augments the tensor created for each domain by incorporating semantic entities behind the sparse topics into the tensor. The set of sparse topics $\mathbb{T}^d$ in domain d is defined as the group of the most sparsely observed topics t_s^ds among all topics in domain d. It is computed as follows.

1) CrTF first sorts the topics in domain d from the rarest to the most common and creates a list of topics: $\{t_{s(1)}^d, t_{s(2)}^d, \ldots, t_{s(n-1)}^d, t_{s(n)}^d\}$. For example, $t_{s(2)}^d$ is not less sparsely observed than $t_{s(1)}^d$.
2) It iterates the step 3 from $j = 1$ to $j = N$.
3) If it satisfies the following equation, CrTF adds the jth sparse item $t_{s(j)}^d$ to set $\mathbb{T}^d$: $(|\mathbb{T}^d|/\sum_{m,n,k} x_{m,n,k}) < \delta$, where $\mathbb{T}^d$ initially does not have any items and $|\mathbb{T}^d|$ is the number of items in set $\mathbb{T}^d$. If not, it stops the iterations and returns the set $\mathbb{T}^d$ as the most sparsely observed topics. Here, $x_{m,n,k}^d$ is 1 if $r_{m,n,k}^d$ is observed and 0 if unobserved.

In the above-mentioned procedure, δ is a parameter used to determine the number of sparse topics in $\mathbb{T}^d$.

We denote an entity linked to a topic t_j^d as e_j. Then, we can denote the number (size) of neighboring entities of e_j, which is linked by the sparse topic t_j^d, as $S^d = |\bigcup_{t_j^d \in \mathbb{T}^d} f(e_j)|$, where the function $f(e_j)$ returns the set of neighboring entities of e_j in the DBpedia network. We also denote the sth neighboring entity in set $\bigcup_{t_j^d \in \mathbb{T}^d} f(e_j)$ as ne_s. Hereafter, for simplicity, this paper assumes the number of domains is two.

CrTF constructs an augmented tensor for domain d, $\mathcal{A}^d$, as follows.

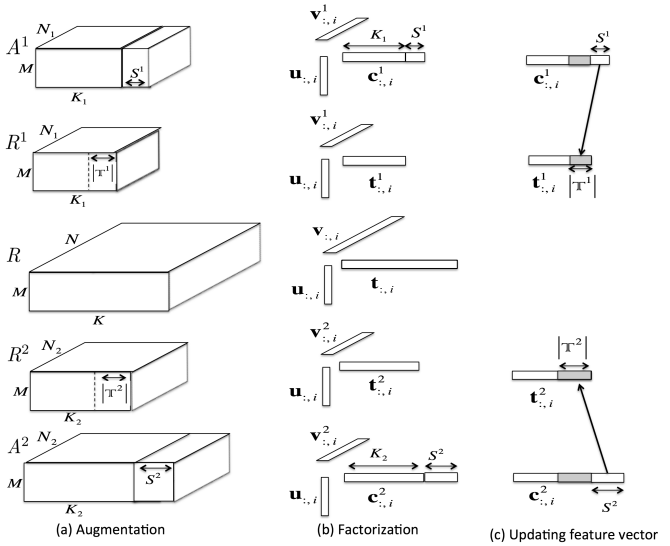


Fig. 2. Examples of our factorization process.

1) CrTF inserts each observed element $r^d_{m,n,j}$ into $\mathcal{A}^d$ as $a^d_{m,n,j}$ if j is $(1 \leq j \leq K)$. For example, let us consider the augmentation of the tensor $\mathcal{R}^1$ in domain 1 in Fig. 1. CrTF inserts multiobject relationships composed of u_1, t_2, and v_1 in $\mathcal{R}^1$ into $\mathcal{A}^1$.
2) CrTF also inserts multiobject relationship composed of user u_m, retweeted user v_n, and neighboring entity ne_s of sparse topics across domains into $\mathcal{A}^d$. This element, $a^d_{m,n,(K+s)}$, is computed as follows $(1 \leq s \leq S^d)$:

$$a^d_{m,n,(K+s)} = \sum_{1 \leq l \leq 2, e_i \in f(ne_s)} a^l_{m,n,i} \Big/ \sum_{1 \leq l \leq 2, e_i \in f(ne_s)} x^l_{m,n,i}.$$

In the above-mentioned equation, the reason why it computes the average value of $a^l_{m,n,i}$ is that it avoids inserting duplicated relationships made by the same combination of user u_m and retweeted user v_n. For example, in Fig. 1, CrTF inserts multiobject relationships composed of u_1, E_2, and v_1 in $\mathcal{R}^1$ into $\mathcal{A}^1$, if t_1 is a sparse topic in domain 1. It also inserts multiobject relationships composed of u_2, E_5, and v_2 in $\mathcal{R}^1$ into $\mathcal{A}^1$, if t_3 is sparse topic in domain 2.

In Fig. 2(a), CrTF propagates the sparse topics in $\mathbb{T}^1$ (or $\mathbb{T}^2$) to neighboring entities [the size is S^1 (S^2)] and creates an augmented tensor $\mathcal{A}^1$ ($\mathcal{A}^2$) from $\mathcal{R}^1$ ($\mathcal{R}^2$).

3) Simultaneously Factorizing Tensors Across Domains: CrTF now factorizes tensors across domains simultaneously by using the VNTF framework. It extends the ideas of SRTF to perform CrTF. The key ideas in our factorization process are as follows.

1) It simultaneously factorizes the original tensor $\mathcal{R}$, tensors $\mathcal{R}^d$s, and their augmented tensors $\mathcal{A}^d$s. During these factorizations, it computes the feature vector $\mathbf{u}_m$ from $\mathcal{R}$, the feature vectors $\mathbf{v}^d_n$ and $\mathbf{t}^d_k$ from $\mathcal{R}^d$, and the semantically biased feature vector for topics $\mathbf{c}^d_j$ from $\mathcal{A}^d$. Here, $\mathbf{v}^d_n$s are shared in the factorizations of $\mathcal{R}^d$ and $\mathcal{A}^d$. As a result, the semantic biases from $\mathcal{A}^d$ can be shared among the tensor factorizations via those parameters. Fig. 2(b) shows the ith row vector of the feature vectors factorized from the tensors. For example, $\mathbf{v}^1_{:,i}$ is shared in factorizing $\mathcal{R}^1$ and $\mathcal{A}^1$.
2) It also lets the feature vector for each user $\mathbf{u}_m$, which is computed by factorizing original tensor $\mathcal{R}$, be shared in the factorizations of $\mathcal{R}^1$, $\mathcal{R}^2$, $\mathcal{A}^1$, and $\mathcal{A}^2$. This is because users are included in multiple domains, and thus, their feature vectors should be learned from multiobject relationships in both domains. This approach has another effect; it also can circulate semantic biases learned from $\mathcal{A}^1$ and $\mathcal{A}^2$ across domains via the user feature vector. In Fig. 2(b), the factorizations of tensor $\mathcal{R}$, $\mathcal{R}^1$, $\mathcal{R}^2$, $\mathcal{A}^1$, and $\mathcal{A}^2$ share $\mathbf{u}_{:,i}$.
3) It updates the latent feature for the sparse topic $\mathbf{t}^d_s$ by incorporating semantic biases from $\mathbf{c}^d_j$s. $\mathbf{c}^d_j$s are feature vectors for neighboring entities of the entity linked by t^d_s. In Fig. 2(c), each row vector $\mathbf{c}^d_{:,i}$ has latent features for K_d topics and those for S^d entities. The features in $\mathbf{c}^d_{:,i}$ share semantic knowledge of the sparse topics and are helpful to solve the sparsity problem.

4) CrTF Procedure: The procedure is summarized as below following the way we did for VNTF.

1) Create the original tensor $\mathcal{R}$, the tensors for the individual domains $\mathcal{R}^d$, and their augmented tensors $\mathcal{A}^d$.
2) Initialize a latent feature matrix for users $\mathbf{U}$ for the original tensor as well as a feature matrix for retweeted users $\mathbf{V}^d$, one for topics $\mathbf{T}^d$, and a semantically biased feature matrix for topics $\mathbf{C}^d$ in each domain d in the same way as the VNTF procedure [e.g., $\mathbf{C}^{d(0)} \sim g(A^d_c, B^d_c./A^d_c)$]. Similarly, initialize a logarithmic feature matrices for users $\mathbf{U}'$ for the original tensor as well as a logarithmic feature matrix for retweeted users $\mathbf{V}^{d'}$, one for topics $\mathbf{T}^{d'}$, and a semantically biased logarithmic feature matrix for topics $\mathbf{C}^{d'}$ in each domain d in the same way as the VNTF procedure [e.g., $\mathbf{C}^{d'(0)} \sim g(A^d_c, B^d_c./A^d_c)$].
3) Repeat the following steps L times, where L is the maximum iteration count.
   a) Repeat steps (a)–(e) in the order of d $(1 \leq d \leq 2)$. During each iteration, $\mathbf{U}$ is shared among the tensor factorizations in two domains [approach (B)].
      i) Compute sources of sufficient statistics for $\mathbf{U}$, $\mathbf{V}^d$, $\mathbf{T}^d$, and $\mathbf{C}^d$, denoted as Σ_u, Σ^d_v, Σ^d_t, and Σ^d_c, respectively. The computation proceeds in the same way as step 2(a) of the VNTF procedure. For example, Σ^d_t is computed as

$$\Sigma^d_t = \frac{1}{2}\mathbf{T}^{d'} .* \left(\left(\sum_m (\mathbf{X}\mathbf{R}^d_m ./ (\mathbf{T}^{d'}\mathbf{V}^{d'}))\mathbf{V}^{d'T} \right) + \left(\sum_n (\mathbf{X}\mathbf{R}^d_n ./ (\mathbf{T}^{d'}\mathbf{U}'))\mathbf{U}'^T \right) \right) \quad (6)$$

and Σ_c^d is computed as

$$\Sigma_c^d = \frac{1}{2}\mathbf{c}^{d'} .* \left(\left(\sum_m (\mathbf{XR}_m^d \ ./ (\mathbf{c}^{d'}\mathbf{V}^{d'}))\mathbf{V}^{d'T}\right) + \left(\sum_n (\mathbf{XR}_n^d \ ./ (\mathbf{c}^{d'}\mathbf{U}'))\mathbf{U}'^T\right)\right) \tag{7}$$

where $\mathbf{XR}_m^d \equiv \mathbf{X}_{m,:,:}^d .* \mathbf{R}_{m,:,:}^d$ and $\mathbf{XR}_n^d \equiv \mathbf{X}_{:,n,:}^d .* \mathbf{R}_{:,n,:}^d$. The above-mentioned two equations share $\mathbf{U}'$ and $\mathbf{V}^{d'}$ [approach (A)].

ii) Update the feature matrices $\mathbf{U}$, $\mathbf{V}^d$, $\mathbf{T}^d$, and $\mathbf{C}^d$. The update proceeds in the same way as step 2(b) in the VNTF procedure. For example, $\mathbf{T}^d$ is computed as

$$\mathbf{T}^d = \alpha_t^d .* \beta_t^d$$

where $\alpha_t^d = A_t^d + \Sigma_t^d$ and $\beta_t^d = 1./(A_t^d./B_t^d + (1/2)\sum_k(\mathbf{R}_{:,:,k}^d\mathbf{U}^T + \mathbf{R}_{:,:,k}^{dT}\mathbf{V}^d))$.
Similarly, $\mathbf{C}^d$ is computed as

$$\mathbf{C}^d = \alpha_c^d .* \beta_c^d$$

where $\alpha_c^d = A_c^d + \Sigma_c^d$ and $\beta_c^d = 1./(A_c^d./B_c^d + \frac{1}{2}\sum_k(\mathbf{R}_{:,:,k}^d\mathbf{U}^T + \mathbf{R}_{:,:,k}^{dT}\mathbf{V}^d))$. As in step (a) [approach (A)], the above-mentioned two equations share $\mathbf{U}$ and $\mathbf{V}^d$.

iii) Update $\mathbf{t}_k^d$ by incorporating semantic biases from $\mathbf{c}_j^d$s in $\mathbf{t}_k^d$ if $t_k^d \in \mathbb{T}^d$ as following equation [approach (C)]:

$$\mathbf{t}_k^d = \frac{\sum_{e_j \in f(e_k)} \mathbf{c}_{(K+j)}^d}{|f(e_k)|}.$$

iv) Update $\mathbf{U}'$, $\mathbf{V}^{d'}$, $\mathbf{T}^{d'}$, and $\mathbf{C}^{d'}$ in the same way as VNTF does [e.g., $\mathbf{C}^{d'(l+1)} = \exp(\Psi(\alpha_c^d)) .* \beta_c^d$].

v) Update $\mathbf{t}_k^{d'}$ by incorporating $\mathbf{c}'^d$s in $\mathbf{t}_k^{d'}$ as in step (c) [approach (C)].

4) Finally, CrTF computes unobserved elements in $\mathcal{R}^d$ by substituting $\mathbf{U}$, $\mathbf{V}^d$, and $\mathbf{T}^d$ in (3).

The computational complexity of CrTF is linear with respect to that of VNMF, which is $(3(M+N+K)+(S_1+S_2))$. This is, however, not a barrier to real applications, because a scalable VNMF has been proposed [10].

V. Evaluation

This paper evaluates CrTF from the following viewpoints: 1) the accuracies of communication frequencies and 2) the results of clustering key topics and social media influencers with semantic labels.

A. Data Set

We use the following two data sets[3]:

[3]They were crawled in the COSMIC project: http://tw.rpi.edu/web/project/cosmic

TABLE II
Examples of Entities Used in the Indian Data Set (They Are Separated by Commas in This Table)

Category:14th_Lok_Sabha_members, Category:Indian_activists, Haryana, Category:Indian_revolutionaries, AlumniOfTrinityCollege, Gandhinagar

1) The *Indian* data set includes tweets for the topic domains, "Rahul Gandhi" and "Narendra Modi," the political leaders of the INC and BJP. The tweets were from January to February in 2014. Users discussing these groups formed different social networks. The INC data set had 345 094 tweets, 6429 retweeted users, 1219 topics linked to DBpedia entities, and retweets by 25 549 users. The BJP one had 325 906 tweets, 6919 retweeted users, 2166 topics linked to DBpedia entities, and retweets by 55 386 users; 6283 users were in both networks. Table II shows examples of DBpedia entities extracted from this data set. The INC data set and the BJP data set have almost the same number of tweets; however, the number of users, that of topics, and that of retweeted users in the INC data set are much more than those in BJP data set. Thus, the communication frequencies made in the INC data set are much more greater than those in the BJP data set.
2) The *Middle East* data set that includes tweets for the Hamas and Hezbolla political topic domains. The tweets were from April to May in 2013. The Hamas data set had 43 015 tweets, 5484 retweeted users, 768 topics linked to DBpedia entities, and retweets by 5166 users. The Hezbolla one had 123 670 tweets, 8477 retweeted users, 1063 topics linked to DBpedia entities, and retweets by 20 610 users; 2102 users were in both networks. Apparently, the communication frequencies made in the Hezbolla data set are much more greater than those in Hamas data set.

B. Compared Methods

The compared methods were as follows.

1) *VNTF* factorizes the original tensor.
2) *SRTF* factorizes the original tensor and its augmented tensor.
3) *Cr-GCTF* applies GCTF [7] to our Idea and factorizes tensors for individual domains simultaneously. GCTF is the current best method that factorizes tensors simultaneously to predict observation frequencies that follow the Poisson–Gamma distribution. It, however, cannot use semantics from DBpedia.
4) *CrTF(1.0)* factorizes the original tensor, tensors for individual domains, and their augmented tensors simultaneously while applying semantics to all topics ($\delta = 1.0$).
5) *CrTF(0.5)* (our method) factorizes the tensors simultaneously while applying semantics to sparse topics ($\delta = 0.5$).

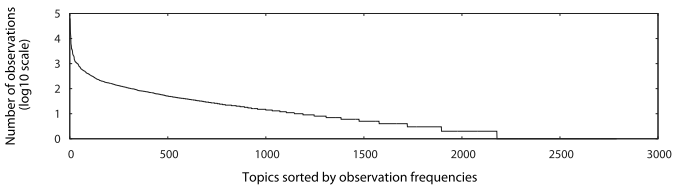


Fig. 3. Distribution of topic frequencies.

The previous social network analyses using matrix factorization [6], [28] focus on predicting relationships composed of two objects. Those that use the random walk method [8] focus on ranking objects. Thus, they cannot be compared with our method.

C. Methodology

We split the Indian (Middle East) data set into retweets in January (April) and those in February (May). We used the retweets in the first month as the training data set and those in the last month as the test data set. Following the methodology used in previous studies [5], [10], [17], we computed the root mean square error (RMSE), i.e., $((\sum_{i=1}^{n}(P_i - R_i)^2)/n)^{1/2}$, where n is the number of entries in the test data set, and P_i and R_i are the predicted and actual values of the ith entry, respectively. To assess the overall improvement in accuracy, the RMSE values were computed for two domains jointly. We set the number of feature vectors, D, to 10, since it gave the highest accuracies for all methods. Following [5], the elements in A_x^d and B_x^d, which are used to generate Gamma distributions, were 1 ($x \in \{u, v, t, c\}$). Iteration count, L, used in VNMF, VNTF, and CrTF was set to 100.

D. Results

We first investigated the sparseness of the topics. Fig. 3 shows the log10 scale distribution of topic frequencies in the Indian data set. We can see that the frequencies exhibit a long-tail characteristic. Thus, the number of observed multiobject relationships became very sparse relative to the possible combinations of topics. The distribution of the Middle East data set showed a similar tendency.

We then confirmed the accuracy of *CrTF(0.5)* when we varied the number of hops h to determine the size of the domains from one to three. The accuracy when h equaled 2 was better than that when h equaled 1, because the observed topics were too sparse to compute predictions when h equaled 1. Accuracy became worse when h equaled 3, because the bias of popular topics grows as the social domain becomes wider. Thus, we set h to 2 hereafter.

We also confirmed the accuracy of *CrTF* saturated before $L = 50$. This means that *CrTF* converges quickly. Because *CrTF* shared the feature matrices among the tensors in different domains, the RMSE values changed dramatically in the first few iterations and saturated rapidly.

Tables III and IV compare the accuracies of the methods on both data sets. Here, *CrTF(0.5)* and *SRTF* performed much better (lower RMSE) than *VNTF*. This is because *CrTF(0.5)* can avoid a strong bias on one of the two domains and semantic biases are helpful for solving the sparsity problem in tensor factorization. On the other hand, *VNTF* produced poor predictions, because the BJP (Hamas) users are more numerous than the INC (Hezbolla) users. *SRTF*, however, was less accurate than *CrTF(0.5)*. Thus, we can see that the coupled analysis of two tensors is useful, because it can circulate the knowledge of the communication tendencies of users as well as semantic biases from the shared semantic concept space across domains. Furthermore, *CrTF(0.5)* outperformed *Cr-GCTF*. Note that the accuracy of *Cr-GCTF* owes to our idea that avoids the bias problem toward some of the domains, which have more observations than others by creating tensors for individual domains and factorizing them simultaneously. GCTF is the current best tensor method that factorizes several tensors simultaneously according to the evaluation made by [7]. Thus, from these results, we can conclude that our factorization method that incorporates semantic biases into the factorizations of tensors created for individual domain is very useful. *CrTF(0.5)* was better than *CrTF(1.0)* though *CrTF(1.0)* is still better than *Cr-GCTF*. This indicates that it is not useful to incorporate semantics in nonsparse objects. Finally, *CrTF(0.5)* was more accurate than the other methods. It achieves higher accuracy than the state-of-the-art methods, *VNTF* and *SRTF*, with a statistical significance of $\alpha < 0.05$.

TABLE III
RMSE VALUES FOR INDIAN DATA SET

	VNTF	*SRTF*	*Cr-GCTF*	*CrTF(1.0)*	*CrTF(0.5)*
$D = 10$	2.9984	2.9712	2.9415	2.9260	***2.9182***
$D = 20$	3.0192	2.9727	2.9552	2.9301	2.9277

TABLE IV
RMSE VALUES FOR MIDDLE EAST DATA SET

	VNTF	*SRTF*	*Cr-GCTF*	*CrTF(1.0)*	*CrTF(0.5)*
$D = 10$	1.8221	1.7982	1.7724	1.7441	***1.7417***
$D = 20$	1.8872	1.8024	1.7914	1.7942	1.7727

Table V shows examples of differences between the predictions of *VNTF* and *CrTF(0.5)* for the Indian data set. The columns "*VNTF*," "*CrTF*," and "Actual" list predictions by *VNTF*, *CrTF(0.5)*, and actual values for the multiobject relationships found in the test data set, respectively. *CrTF(0.5)* could more accurately predict multiobject relationships composed of users, topics, and retweeted users than *VNTF* could, since it could use the semantics behind the topics being discussed. For example, a tweet including "Mohan Bhagwat" [the chief of the Rashtriya Swayamsevak Sangh (RSS)] described by retweeted-user "2" was retweeted by user "1" in the training data set. In addition, a tweet including "RSS"[4] described by retweeted-user "2" was retweeted by user "1" in the test data set. Our data set did not include many tweets including "RSS" or "Mohan Bhagwat" and those DBpedia entities are included in BJP domain in our data set; however, *CrTF(0.5)* accurately predicted this combination, since it could use the

[4]The right-wing charitable, educational, volunteer, Hindu nationalist, and nongovernmental organization.

TABLE V
EXAMPLES OF PREDICTIONS MADE BY *VNTF*/*CrTF(0.5)* FOR INDIAN DATA SET

Training dataset				Predictions by *VNTF* and *CrTF(0.5)*					
User	Topic	Retweeted user	Frequency	User	Topic	Retweeted user	*VNTF*	*CrTF*	Actual
1	Mohan Bhagwat	2	1	1	Rashtriya Swayamsevak Sangh	2	1.5	1.7	2
3	Manmohan Singh	4	2	3	Atal Bihari Vajpaye	4	0.7	0.9	1

TABLE VI
CLUSTERING MULTIOBJECT RELATIONSHIPS FOR INDIAN DATA SET

Cluster	User group	Topics in the cluster with DBpedia semantic knowledge	Retweeted user group
A	1	Raj Babbar (Political party/INC), Jawaharlal Nehru (Political party/INC), Sandeep Dikshit (Political party/INC, Residence/New Delhi), Gita Mehta (Occupation/Author, Documentary, Filmmaker, Journalist), Lok Sahba (Meeting place/New Delhi)	2
B	3	Rashtriya Swayamsevak Sangh (Type/Voluntary,Paramilitary), Lalu Prasad Yadav (Political party/RJD), Bal Thackeray (Political party/Shiv Sena), Non-resident person of Indian origin (Related ethnic groups/Indian people), Peter Hain (Political party/Labour)	4

semantic knowledge that "Mohan Bhagwat" is a neighbor entity of "RSS" in the DBpedia knowledge base ("Mohan Bhagwa" has a predicate "keyPerson" with "RSS" in DBpedia) as well as it could avoid a strong bias on the INC data set.

Another example is that a tweet including "Manmohan Singh"[5] by retweeted-user "4" was retweeted by user "3" in the training data set. Furthermore, a tweet including "Atal Bihari Vajpayee"[6] by retweeted-user "4" was retweeted by user "3" in the test data set. Our data set did not have many tweets including "Atal Bihari Vajpaye" and "Atal Bihari Vajpaye" is included in the BJP domain; however, *CrTF(0.5)* accurately predicts this combination, since it could avoid a strong bias on the INC data set and it could use the semantic knowledge that "Manmohan Singh" has a predicate "successor" (of Prime minister) with "Atal Bihari Vajpaye." Note that CrTF has other applications. For example, if we use music topics instead of political ones, a communication prediction for music can be computed. This prediction can be applied to music recommendation systems that now use users' social relationships in their recommendations.

Table VI shows the clustering results for the Indian data set. We computed the probability that the multiobject relationship composed of u_m, v_n, and t_k is included in the ith cluster among D dimensions by the following equation:

$$u_{i,m} \cdot v_{i,n} \cdot t_{i,k} \quad (1 \leq i \leq D).$$

We selected the cluster that gave the highest value for the above-mentioned equation for each multiobject relationship. The columns "User group," "Topics," and "Retweeted user group" present the user groups, topics, and retweeted user groups classified in the same dimension, respectively. The column "Topics" also refers to the combination of the DBpedia predicate and the subject (or object) of each entity. Thus, the table presents key topics and social media influencers with semantic labels. By checking the DBpedia knowledge, we can easily understand the background of each cluster. In each cluster, we can also find the retweeted user group that provided the authoritative tweets for the topics in the cluster as well as the user group that distributed those topics. For example, cluster A had user group 1 (that distributed topics focusing on political leaders in "INC") as described by retweeted user group 2. It nicely contained the related topics such as "Raj Babbar," a politician in "INC," "Gita Metha" who is a daughter of a famous INC politician "Biju Patnaik," and "Lok Sahba," which is the lower house of the Parliament of India. In the same way, cluster B contained user group 3 (that distributed topics focusing on political leaders in "Rashtriya Janata Dal" and "Shiv Sena" that belong to the "United Progressive Alliance" as well as international political experts on Indian politics such as "Peter Hain") as described by retweeted user group 4. Interestingly, cluster B mainly includes the communications made for the third parties of Indian politics. We could not extract such clusters by using *VNTF*. This is because *CrTF* could avoid a strong bias on the INC data set and well extract a group of such communications.

The clustering results let political parties analyze who are social influencers, what topics they are tweeting, and who retweets those influencers' opinions to their followers with respect to those parties. By analyzing the users' descriptions tweeted in those clusters, political parties can reflect those opinions to their policies.[7]

VI. CONCLUSION

This paper proposes CrTF that represents a new research direction for cross domain analysis, since it can effectively use semantic knowledge in LOD shared by multiple service domains. CrTF links topics in tweets to DBpedia entities and creates tensors that represent social networks for corresponding topic domains. It simultaneously factorizes tensors across domains while integrating semantics from DBpedia into the factorization. Experiments showed that CrTF outperforms the current tensor-based methods. We will apply our ideas to the analysis of social networks for multiple domains and

[5]An Indian economist who served as the Prime Minister of India from 2004 to 2014.

[6]An Indian statesman who was the 10th Prime Minister of India, first for 13 days in 1996 and then from 1998 to 2004.

[7]We omit more detailed social opinions described in tweets, since they are private tweets on politics written by citizen users and should not be published in the paper.

semantic entity space other than DBpedia. We also extend our method to more than 3-D problems like time-evolving social networks, where we need to handle a tensor composed by users, retweeted users, topics, and time periods. It will also be interesting to see this approach being applied on other systems (e.g., music service incorporating social network information). Furthermore, it will incorporate the feedback from users across different applications to enhance the algorithm performance.

REFERENCES

[1] C. Anderson, *The Long Tail: Why the Future of Business is Selling Less of More*. New York, NY, USA: Hyperion, 2006.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. New York, NY, USA: Springer-Verlag, 2006.

[3] C. Bizer *et al.*, "DBpedia—A crystallization point for the Web of data," *Web Semantics*, vol. 7, no. 3, pp. 154–165, 2009.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[5] A. T. Cemgil, "Bayesian inference for nonnegative matrix factorisation models," *Intell. Neurosci.*, vol. 2009, pp. 4-1–4-17, Feb. 2009.

[6] D. M. Dunlavy, G. T. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations," *ACM Trans. Knowl. Discov. Data*, vol. 5, no. 2, pp. 10-1–10-27, 2011.

[7] B. Ermiş, E. Acar, and A. T. Cemgil, "Link prediction in heterogeneous data via generalized coupled tensor factorization," *Data Mining Knowl. Discovery*, vol. 29, no. 1, pp. 203–236, Dec. 2015.

[8] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme, "Information retrieval in folksonomies: Search and ranking," in *Proc. ESWC*, 2006, pp. 411–426.

[9] M. Jiang *et al.*, "Social contextual recommendation," in *Proc. CIKM*, 2012, pp. 45–54.

[10] Y.-D. Kim and S. Choi, "Scalable variational Bayesian matrix factorization with side information," in *Proc. AISTATS*, 2014, pp. 493–502.

[11] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[12] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher, "MetaFac: Community discovery via relational hypergraph factorization," in *Proc. KDD*, 2009, pp. 527–536.

[13] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: Social recommendation using probabilistic matrix factorization," in *Proc. CIKM*, 2008, pp. 931–940.

[14] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "DBpedia spotlight: Shedding light on the Web of documents," in *Proc. I-Semantics*, 2011, pp. 1–8.

[15] M. Nakatsuji, M. Yoshida, and T. Ishida, "Detecting innovative topics based on user-interest ontology," *Web Semantics*, vol. 7, no. 2, pp. 107–120, 2009.

[16] M. Nakatsuji, Y. Fujiwara, A. Tanaka, T. Uchiyama, and T. Ishida, "Recommendations over domain specific user graphs," in *Proc. ECAI*, 2010, pp. 607–612.

[17] M. Nakatsuji, Y. Fujiwara, H. Toda, H. Sawada, J. Zheng, and J. A. Hendler, "Semantic data representation for improving tensor factorization," in *Proc. AAAI*, 2014, pp. 2004–2012.

[18] M. Nakatsuji, H. Toda, H. Sawada, J. G. Zheng, and A. J. Hendler, "Semantic sensitive tensor factorization," *Artif. Intell.*, vol. 230, pp. 224–245, Jan. 2016.

[19] M. Nakatsuji, "Semantic sensitive simultaneous tensor factorization," in *Proc. Int. Semantic Web Conf.*, vol. 9981. 2016, pp. 411–427.

[20] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima, "Tensor factorization using auxiliary information," in *Proc. ECML-PKDD*, 2011, pp. 501–516.

[21] M. Nasim, R. Charbey, C. Prieur, and U. Brandes, "Investigating link inference in partially observable networks: Friendship ties and interaction," *IEEE Trans. Comput. Social Syst.*, vol. 3, no. 3, pp. 113–119, Sep. 2016.

[22] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proc. WSDM*, 2010, pp. 81–90.

[23] K. Takeuchi, R. Tomioka, K. Ishiguro, A. Kimura, and H. Sawada, "Non-negative multiple tensor factorization," in *Proc. ICDM*, 2013, pp. 1199–1204.

[24] F.-Y. Wang *et al.*, "Where does AlphaGo go: From church-turing thesis to AlphaGo thesis and beyond," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 2, pp. 113–120, Apr. 2016.

[25] Y. K. Yilmaz, A.-T. Cemgil, and U. Simsekli, "Generalised coupled tensor factorisation," in *Proc. NIPS*, 2011, pp. 2151–2159.

[26] W. Zhang, T. J. Johnson, T. Seltzer, and S. L. Bichard, "The revolution will be networked," *Soc. Sci. Comput. Rev.*, vol. 28, no. 1, pp. 75–92, 2010.

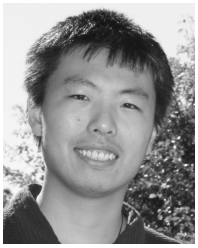
[27] Q. Zhang, D. D. Zeng, F. Y. Wang, R. Breiger, and J. A. Hendler, "Brokers or bridges? Exploring structural holes in a crowdsourcing system," *Computer*, vol. 49, no. 6, pp. 56–64, Jun. 2016.

[28] Y. Zhen, W.-J. Li, and D.-Y. Yeung, "TagiCoFi: Tag informed collaborative filtering," in *Proc. RecSys*, 2009, pp. 69–76.

[29] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, "Collaborative filtering meets mobile recommendation: A user -centered approach," in *Proc. AAAI*, 2010, pp. 236–241.

Makoto Nakatsuji received the B.S. degree in applied mathematics and physics and the M.S. degree from the Department of Systems Science, Kyoto University, Kyoto, Japan, and the Ph.D. degree in social informatics from the Kyoto University Graduate School of Informatics, Kyoto, in 2010.

He is currently a Manager of NTT Resonant Inc., Tokyo, Japan. He was a Visiting Scholar with the The Tetherless World Constellation, Rensselaer Polytechnic Institute, Troy, NY, USA, in 2013. His research interests include question–answering systems, deep learning algorithms, semantic data mining, recommendation, and link prediction.



Qingpeng Zhang (M'10) received the B.S. degree in automation from the Huazhong University of Science and Technology, Wuhan, China, and the M.S. degree in industrial engineering and the Ph.D. degree in systems and industrial engineering from The University of Arizona, Tucson, AZ, USA.

He was a Post-Doctoral Research Associate with The Tetherless World Constellation, Rensselaer Polytechnic Institute, Troy, NY, USA. He is currently an Assistant Professor with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong. His research interests include social computing, complex networks, healthcare data analytics, semantic social networks, and Web science.

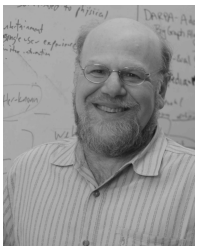
Dr. Zhang is currently an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

Xiaohui Lu received the B.S. degree in computer science from University at Albany (SUNY), Albany, NY, USA, and the M.S. and Ph.D. degrees in computer science from the Rensselaer Polytechnic Institute, Troy, NY, USA.

He was a Post-Doctoral Research Associate with Tetherless World Constellation, Rensselaer Polytechnic Institute. He is currently a Software Engineer with Expedia. His research interests include computational social science, semantic Web, data mining, and machine learning.

Bassem Makni received the master's degree from the National School of Computer Science, Manouba, Tunisia, and the M.Phil. degree from the KMI Group, Open University, Milton Keynes, U.K., where he was involved on the semantic Web services. He is currently pursuing the Ph.D. degree with the Rensselaer Polytechnic Institute, Troy, NY, USA, under the supervision of Prof. J. Hendler on deep learning for noise-tolerant semantic reasoning.

He held internships with INRIA, Sophia Antipolis, France, where his research interests focused on building ontologies from emails corpus and ontology-based navigation.



James A. Hendler (F'10) is currently the Director with the Institute for Data Exploration and Applications and the Tetherless World Professor of computer, Web, and cognitive sciences with the Rensselaer Polytechnic Institute, Troy, NY, USA. He has authored over 350 books, technical papers, and articles in the areas of semantic Web, artificial intelligence, agent-based computing, and high performance processing.

Prof. Hendler is a fellow of the American Association for Artificial Intelligence, the British Computer Society, and the AAAS. One of the originators of the Semantic Web, he was a recipient of a 1995 Fulbright Foundation Fellowship. He is a former member of the U.S. Air Force Science Advisory Board. He is also the former Chief Scientist of the Information Systems Office, U.S. Defense Advanced Research Projects Agency. He received U.S. Air Force Exceptional Civilian Service Medal in 2002. In 2010, He was named one of the 20 most innovative professors in America by *Playboy Magazine* and was selected as an Internet Web Expert by the U.S. Government. In 2012, he was one of the inaugural recipients of the Strata Conference Big Data Awards for his work on large-scale open government data, and he is a Columnist and an Associate Editor of the *Big Data* journal. In 2013, he was appointed by the Governor as the Open Data Advisor to NY. He is also the first Computer Scientist to serve on the board of *Reviewing Editors* for science.