Labeling Actors in Multi-view Social Networks by Integrating Information From Within and Across Multiple Views

Ngot Bui
College of IST
Pennsylvania State University
University Park, PA 16802, USA
npb123@ist.psu.edu

Thanh Le
College of IST
Pennsylvania State University
University Park, PA 16802, USA
txl252@ist.psu.edu

Vasant Honavar
College of IST
Pennsylvania State University
University Park, PA 16802, USA
vhonavar@ist.psu.edu

***Abstract*—Real world social networks typically consist of actors (individuals) that are linked to other actors or different types of objects via links of multiple types. Different types of relationships induce different *views* of the underlying social network. We consider the problem of labeling actors in such multi-view networks based on the connections among them. Given a social network in which only a subset of the actors are labeled, our goal is to predict the labels of the rest of the actors. We introduce a new random walk kernel, namely the Inter-Graph Random Walk Kernel (IRWK), for labeling actors in multi-view social networks. IRWK combines information from within each of the views as well as the links across different views. The results of our experiments on two real-world multi-view social networks show that: (i) IRWK classifiers outperform or are competitive with several state-of-the-art methods for labeling actors in a social network; (ii) IRWKs are robust with respect to different choices of user-specified parameters; and (iii) IRWK kernel computation converges very fast within a few iterations.**

***Index Terms*—Multi-view Network, Inter-Graph Random Walk Kernel; Social Network Analysis**

I. Background and Introduction

A. Background

The emergence and wide adoption of social networks e.g., Facebook[1], Google+[2] and social media (e.g., Youtube[3]) offer a rich source of *big data* for understanding the structure, dynamics, formation and evolution of social structures, as well as the role of such networks in influencing individual and collective behavior, e.g., interest in particular products, affiliation in specific groups, or participation in specific activities, etc. Real world social networks are characterized by multi-dimensional social relationships among individuals or actors. Each actor can link to multiple other actors; and each actor can have multiple types of relationships with a given actor. For example, some, but not all of one's co-workers may also be one's friends. The relationships between actors can be of different types, e.g., friendship, co-authorship, etc., [1], [2], [3]. For example, Google+ allows members to specify different 'circles' that correspond to social relationships along different dimensions e.g., friendship, family membership, etc. Similarly, in the DBLP[4] data, each actor (author) can be linked to multiple other authors through a variety of relationships, e.g., co-authorship of articles, publication venues, institutional affiliations, etc. Furthermore, in general, actors in social networks can belong to multiple groups, e.g., authors, musicians, professors, students, etc. Hence, real-world social networks are naturally represented as *multi-view* networks wherein different types of links e.g., friendship, family membership, etc., constitute different *network views* [4], [5], [6].

There is a growing interest in the problem of labeling actors in social networks [7], [8], [9], [10], [11] according to their e.g., political affiliation, interest in particular products, or participation in activities, etc. Actor labels can be used, among other things, for targeting advertisements, recommending products, suggesting membership in social or professional groups, analyzing the social or demographic characteristics of the populations involved, etc. In such a setting, given a social network in which only some of the actors are labeled, the goal is to label the rest of the actors. More precisely, given a social network represented by a graph $G = (V, E)$ in which each actor $x \in V$ belongs to one of C categories in $Y = \{y_1, y_2, \cdots, y_C\}$ and a subset $V^{'}$ of labeled actors, the goal is to label the rest of the actors in $V^U = V - V^{'}$.

B. Related Work

Early work on labeling nodes in networks focused on networks with a single type of nodes and a single type of links. This work has led to methods that exploit correlations among the labels and attributes of nodes [9], [8], [12], [13]; methods, e.g., relational learners, that label an actor by (iteratively) assigning to a node a label that matches the labels of a majority of its neighbors [14]; supervised and or semi-supervised learning methods [15], [16] including those that exploit abstraction hierarchies over nodes to cope with data sparsity [13]; and

[1] https://www.facebook.com/
[2] https://plus.google.com/
[3] https://www.youtube.com/
[4] http://www.informatik.uni-trier.de/~ley/db

kernel based methods [17], [18], [7], [19], [20] that assign similar labels to similar actors (where similarity is defined by a kernel function); random-walk based methods [21], [22], [11] that assign a label to a node based on the known label(s) of the node(s) reachable via random walk(s) originating at the node.

However, as noted above, real world social networks are naturally modeled as multi-view networks that admit multiple types of nodes (actors) and links (relationships). Hence, there is a growing interest in methods for labeling actors in such multi-view social networks. Of particular interest are methods that effectively make use of both the information provided by the multiple network views as well as the interactions between the views. The views in social network can be either explicit or implicit [2]. Eldardiry and Neville [12] proposed an ensemble of relational learners wherein each relational learner focuses on a single social dimension or view of the network. EdgeCluster [1] and SCRN [23] extract the implicit views in the network and use the actors' social dimensions, i.e., social relations (view affiliation) as features to label actors. Ji et al. [24] integrate ranking of network nodes with classification to improve the labeling of nodes in multi-view networks. Wan et al. [25] use information extracted from sequences of linked nodes of different types to label nodes and to identify nodes for which labels should be obtained through active learning to improve the accuracy of the learned models. Bui et al. [7] introduced a heterogeneous graph kernels, a variant of graph kernels applicable in the setting of heterogeneous networks consisting of multiple types of nodes and links, for labeling actors in such networks. Several authors have proposed latent space models to handle the multiple views in multi-view networks. Examples of such models [26] include hypergraph regularized generative models [27], partially shared latent factor models [28]. Latent space joint model [29] learns a latent space representation of a multi-view network and uses it to label the nodes. Other methods, e.g., DeepWalk [11], LINE [30], and Node2Vec [31], that construct and exploit low-dimensional vector space representations that encode social ties of nodes, have been shown to achieve state-of-the-art performance in labeling nodes in networks.

C. Overview

In this paper, we explore a novel approach to labeling actors in multi-view networks. We represent each actor in a multi-view social network in terms of the actor's *intra-view* and *inter-view* contexts where the intra-view context of an actor in a view represents the direct and indirect neighbors and links between neighbors within that view; and the inter-view context of an actor refers to the social contexts of the actor across all of the views other than the current view. Figure 1 schematically depicts the intra-view and inter-view contexts of the actor x that appears in the views V_1, V_2, V_3.

Based on the multi-view representation of social networks outlined above, we introduce a new random walk kernel, the Inter-Graph Random Walk Kernel (IRWK), to efficiently compute the similarity between of any pair of actors in the network based on the social contexts of the actors from different network views. Inter-Graph Random Walk Kernel models the network based on information provided by each of the views as well as relationships between views. IRWK is then used to learn a discriminative classifier to label actors. Results of experiments on two real-world multi-view social network data sets show that: (i) IRWK classifiers outperform or are competitive with several representative methods in labeling actors in a social network; (ii) IRWKs are robust to different choices of user-specified parameters (i.e., stopping probability, transition probability); and (iii) IRWK computation converges in a few iterations.

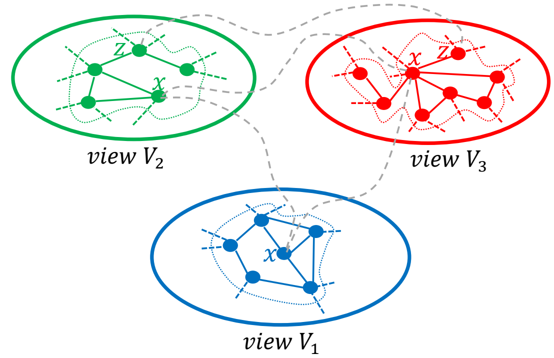


Fig. 1: Multi-view network. Solid lines between nodes within each view denote one type of relationship, e.g., friendship. Dotted lines between nodes in different views encode correspondences between representations of specific actors across different views.

The rest of the paper is organized as follows. Section II describes our approach to integrating information from within and across multiple views of multi-view networks using an inter-graph random walk kernel. Section III presents results of experiments that compare the proposed approach with several representative methods for labeling actors in multi-view social networks. Section IV concludes with a summary and an outline of some promising directions for further research.

II. Integrating Information from Within and Across Views in Multi-View Networks

A. Labeling Actors in Multi-View Networks

As noted earlier, real world social networks can be modeled as multi-view networks wherein each view represents a network structure consisting of single-type of links.

Definition 1: **Multi-view Network.** Let $\mathcal{C} = \{1, 2, \cdots M\}$ be a set of colors (or views, or relation types). Let $V = V_1 \cup V_2 \cup \ldots \cup V_M$, where V_i denotes the set of actors that belong to view i, and $E_{ii} = \{(x, z)\,|x \in V_i, z \in V_i, 1 \leq i \leq M\}$ is a (colored) set of edges that connect actors within view i. A multi-view network consists of a set of graphs $G = \{G_i = (V_i, E_{ii})|1 \leq i \leq M\}$ augmented with inter-view edges $S = \bigcup_{i,j} E_{ij} = \{(u, v)\,|u \in V_i, v \in V_j, i \neq j, i \in \mathcal{C}, j \in \mathcal{C}\}$ is a set of inter-view edges between vertices that belong to different views (Inter-view edges establish correspondences

TABLE I: Symbols & their definitions.

Symbol	Definition
N	Number of actors in a multi-view network
M	Number of views
V_i	Set of vertices of view i.
E_{ij}	Set of edges that link vertices between view i and j
$P_i(z\|x)$	Transition probability from x to z within view i
$P_{i\rightarrow j}(u\|x)$	Inter-view transition probability of x from view i to u in view j where x and u are representations of the same actor.
d	Maximum degree of a node in G
d_v	Maximum degree of node in all view when disregarding inter-view links
$(x,z)_{ij}$	A pair of nodes x,z where $x \in V_i$ and $z \in V_j$.
$\mathbf{T}_{ij}$	Transition probability matrix between nodes from view i to view j
$\mathbf{T}_{\times}$	Transition probability matrix of direct product graph $G_{\times}$

between vertices across different views that represent the same actor)

Definition 2: **Labeling Actors.** Given a multi-view network with a set of actors (vertices) V, where a subset of actors are labeled with labels taken from a label set C, the task is to label the rest of the actors.

B. Random Walk Kernels

Recall that the *random walk kernel* [17], [32] can be used to measure the similarity between pairs of nodes in social networks [7]. Two nodes in a graph are considered to be similar if the sets of random walks originating from the two nodes are similar. To more precisely define the random walk graph kernel, it is useful to introduce the concept of a direct product graph. Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the direct product graph $G_{\times} = (V_{\times}, E_{\times})$ of G_1 and G_2 is a graph with node set $V_{\times} = \{(v_i, v_k) : v_i \in V_1, v_k \in V_2\}$ and edge set $E_{\times} = \{((v_i, v_k), (v_j, v_l)) : (v_i, v_j) \in E_1 \wedge (v_k, v_l) \in E_2\}$. A random walk on the direct product graph is equivalent to simultaneous random walks on G_1 and G_2. Let p_1 and p_2 denote initial probability distributions over the vertices of G_1 and G_2, respectively. Likewise, q_1 and q_2 are stopping probabilities. Then, the starting and stopping probability distributions of the direct graph product $G_{\times}$ are $p_{\times} = p_1 \otimes p_2$ and $q_{\times} = q_1 \otimes q_2$ where $\otimes$ denotes for Kronecker product. The transition probability matrix of the direct product graph $G_{\times}$ is defined in the same manner: $\mathbf{T}_{\times} = \mathbf{T}_1 \otimes \mathbf{T}_2$ where $\mathbf{T}_1$ and $\mathbf{T}_2$ are the transition probability matrices of G_1 and G_2, respectively.

We now proceed to introduce the inter-graph random walk kernel (IRWK), a variant of the random walk graph kernel [17], [32] for integrating information from within and across multiple views of multi-view networks. Specifically, we formulate a kernel function of any pair of actors in a given network view to measure their similarity based on both the intra-view and inter-view contexts of an actor. The resulting IRWK is then used to build a classifier for labeling actors in multi-view social networks.

C. Inter-Graph Random Walk Kernel

We model the relationships of actors across two different views by introducing a new type of link, the *inter-view* link type among the same actors in the two views. Specifically, if an actor belongs to two views, say i and j, there exists a link that connects the two representations of the actor across the two views (see Figure 1).

We define the transition probability between a node x and its neighbors as being inversely proportional to number of x's neighbors. Let $\mathbf{T}_{ij}$ be the transition probability matrix between nodes in V_i and V_j (note that, in general, $\mathbf{T}_{ij} \neq \mathbf{T}_{ji}$). The sum of the probabilities of transitioning from node x in view i to its neighbors (within the same view or in a different view) should sum to 1.

$$\sum_{z \in \mathcal{N}_i(x)} P_i(z|x) + \sum_{j \neq i}^{M} P_{i \rightarrow j}(u|x) + \rho = 1 \tag{1}$$

where $P_i(z|x)$ is the *intra-view* (view i) transition probability from x to z, $\mathcal{N}_i(x)$ is a set of neighbors of x in view i, $P_{i\rightarrow j}(u|x)$ is the *inter-view* transition probability of x from view i to u in view j where x and u are representations of the same actor, and ρ is the probability of remaining at x (stopping probability) at x in view i.

Let x be a node in a multi-view network G and let $h_x = x - x_1 - x_2 - \cdots - x_l$ be a random walk starting from x with the length of l. The probability of h_x is defined as: $P(h_x) = P_s(x) P_t(x_1|x) P_t(x_2|x_1) \cdots P_t(x_l|x_{l-1}) P_e(x_l)$ where $P_s(x)$ and $P_e(x_l)$ are starting and stopping probabilities of x and x_l, respectively and $P_t(x_i|x_{i-1})$ is the (intra-view or inter-view) transition probability. We assume that $P_s(x) = \tau$ and $P_e(x) = \rho$ for nodes in all views. Kernel function between two nodes $x, z \in V$ is defined as follows:

$$K(x,z) = \sum_{h_x} \sum_{h_z} R(h_x, h_z) P(h_x) P(h_z) \tag{2}$$

where $R(h_x, h_z)$, the similarity between two paths h_x and h_z, is equal to 0 if they are of different lengths; otherwise, $R = \prod_{i=1}^{l} R_0(x_i, z_i)$.

We call $R_0(x_i, z_i)$ the initial kernel value between two nodes $x_i, z_i \in V$. If x_i and z_i belong to same view (say j), then $R_0(x_i, z_i) = R_0^j(x_i, z_i)$; otherwise, $R_0(x_i, z_i) = 0$. $R_0^j(x_i, z_i)$ is defined on the sets of directed neighbors of x_i and z_i as follows.

$$R_0^j(x_i, z_i) = \begin{cases} \frac{\sum_{k=1}^{M} |\mathcal{N}_k(x_i) \cap \mathcal{N}_k(z_i)|}{\sum_{k=1}^{M} |\mathcal{N}_k(x_i) \cup \mathcal{N}_k(z_i)|}, & \text{if } x_i, z_i \in V_j \\ 0 & \text{, otherwise} \end{cases} \tag{3}$$

Based on the definitions of $P(h_x)$ and $R_0(x_i, z_i)$, we derive Equation 2 using nested structure as follows.

$$
\begin{aligned}
K(x,z) = \lim_{L\to\infty} \sum_{l=1}^{L} \sum_{x_1,z_1} & \left(P_s(x) P_t(x_1|x) R_0(x_1,z_1)\right. \\
& \times P_s(z) P_t(z_1|z) \sum_{x_2,z_2} \left(P_t(x_2|x_1) R_0(x_2,z_2)\right. \\
& \times P_t(z_2|z_1) \cdots\cdots \sum_{x_l,z_l} \left(P_t(x_l|x_{l-1}) P_e(x_l)\right. \\
& \times R_0(x_l,z_l) P_t(z_l|z_{l-1}) P_e\ (z_l)) \ldots)) \qquad (4)
\end{aligned}
$$

Hence, we have:

$$K(x,z) = \tau^2 \lim_{L\to\infty} R_L(x,z) \qquad (5)$$

where

$R_L(x,z) = \sum_{l=1}^{L} r_l(x,z)$

$r_l(x,z) = \sum_{x_1,z_1} P_t(x_1|x) R_0(x_1,z_1) P_t(z_1|z) r_{l-1}(x_1,z_1)$

$r_1(x,z) = \sum_{x_1,z_1} P_t(x_1|x) R_0(x_1,z_1) P_t(z_1|z) r_0(x_1,z_1)$

$r_0(x,z) = P_e(x) P_e(z)$

Recall that each view represents a network structure where nodes in the view has the same type of social relationship, e.g., friendship, kinship. Each view can be treated as a network of single type of node and single type of link, e.g., friendship node type, friendship link type. Hence, we focus only on the value of the kernel function applied to a pair of actors within the same view (i.e., "same type") by using intra-view context and inter-view context of an actor.

Computing the value of $K(x,z)$ using Equation 5 is not feasible because random walks can be unbounded in length. We can use a kernel matrix to represent, and efficiently compute, the kernel function evaluations over all pairs of nodes in G. We introduce a modification of direct product graph to the setting of multi-view network and use it to Equation 5 in the matrix form (see Table I for symbols and their definitions).

Lemma 1: Consider a multi-view network represented by a graph $G = \{G_i = (V_i, E_{ii}) | 1 \le i \le M\}$ augmented with inter-view edges $S = \bigcup_{i,j} E_{ij}$. If there are two random walks h_x, h_z that simultaneously start from x, z of the same view and end at x_l, z_l, respectively such that $R(h_x, h_z) > 0$. Then there exists a path in $G_\times = (V_\times, E_\times)$ from (x,z) to (x_l, z_l) where $V_\times = \{(x_m, x_n) : x_m, x_n \in V\}$ and $E_\times = \{((x_k, x_q), (x_m, x_n)) : (x_k, x_m),$ $(x_q, x_n) \in E_{ij} \wedge \{x_k, x_q\} \subset V_i \wedge \{x_m, x_n\} \subset V_j\}$.

Proof. Let $h_x = x - x_1 - x_2 - \cdots - x_l$ and $h_z = z - z_1 - z_2 - \cdots - z_l$ be two random walks in G. Clearly, $p = (x,z) - (x_1,z_1) - \cdots - (x_l,z_l)$ is a random walk from (x,z) to (x_l,z_l) and all vertices in this walk $\in V_\times$. Since $R(h_x,h_z) > 0$, $R_0(x_k,z_k) \neq 0 \wedge 1 \le k \le l$, we obtain that x_k, z_k belong to the same view, say view i. WLOG, consider an edge $((x_k,z_k),(x_{k+1},z_{k+1}))$ in p, likewise, x_{k+1}, z_{k+1} are also in the same view, say view j. Furthermore, (x_k, x_{k+1}) and (z_k, z_{k+1}) are edges in h_x and h_y, respectively. As a result, we have $(x_k, x_{k+1}), (z_k, z_{k+1}) \in E_{ij}$ and $\{x_k, z_k\} \subset V_i$ and $\{x_{k+1}, z_{k+1}\} \subset V_j$ or $((x_k,z_k),(x_{k+1},z_{k+1})) \in E_\times$. □

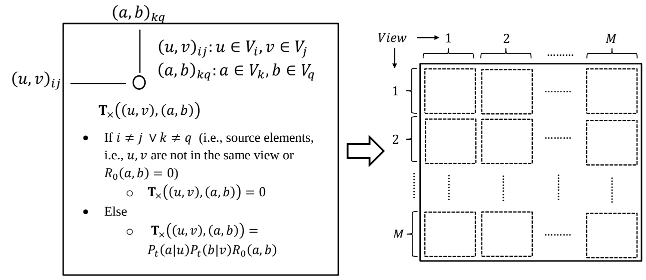


Fig. 2: Reduction on dimension of $\mathbf{T}_\times$

Lemma 1 implies that if two random walks h_x and h_z starting from x and z of the same view are used in computing the kernel value between x and z in Equation 2, there is a corresponding random walk in the modified direct product graph. The formula $\lim_{L\to\infty} R_L(x,z)$ in Equation 5 is written as follows (see Appendix for the derivation).

$$R_\infty(x,z) = r_1(x,z) + \sum_{x_1,z_1} \mathbf{T}_\times((x,z),(x_1,z_1)) R_\infty(x_1,z_1) \qquad (6)$$

where $\mathbf{T}_\times((x,z),(x_1,z_1)) = P_t((x_1,z_1)\,|\,(x,z)) = P_t(x_1|x) R_0(x_1,z_1) P_t(z_1|z)$ is the transition probability from (x,z) to (x_1,z_1) in the modified direct product graph $G_\times$. Using Equation 6, the matrix form of Equation 5 is written as follows (see Appendix for the derivation).

$$\mathbf{K} = \tau^2 (\mathbf{I} - \mathbf{T}_\times)^{-1} \mathbf{r}_1 \qquad (7)$$

where $\mathbf{K} = (K(x_1,x_1), K(x_1,x_2), \cdots, K(x_{MN},x_{MN}))^{\mathrm{T}} \in \mathbb{R}^{M^2N^2}$, $\mathbf{I}$ is an identity matrix, $\mathbf{r}_1 = (r_1(x_1,x_1), r_1(x_1,x_2), \cdots, r_1(x_{MN},x_{MN}))^{\mathrm{T}} \in \mathbb{R}^{M^2N^2}$, and $\mathbf{T}_\times \in \mathbb{R}^{M^2N^2 \times M^2N^2}$.

D. Efficient Computation of IRWK

Computing the kernel value for any pair of vertices $\in V$ is equivalent to the solving linear system in Equation 7. This requires inverting the coefficient matrix $(\mathbf{I} - \mathbf{T}_\times) \in \mathbb{R}^{M^2N^2 \times M^2N^2}$. This coefficient matrix is sparse because the number of non-zero elements of $\mathbf{T}_\times$ is fewer than $(dMN)^2$ (d is the maximum degree of a node in G). Hence, one can deploy efficient numerical algorithms [17] that take advantage of the sparsity of the linear system specified by Equation 7. Specifically, we can use Lemma 1 to reduce the dimensions of $\mathbf{T}_\times$, $\mathbf{K}$, and $\mathbf{r}_1$ to $\mathbb{R}^{MN^2 \times MN^2}$, $\mathbb{R}^{MN^2}$, $\mathbb{R}^{MN^2}$, respectively, taking note of the fact that the value of the kernel function when applied to nodes from different views can be set equal to zero.

Our approach to reducing the dimensionality of $\mathbf{T}_\times$ is shown in Figure 2. Let $\mathbf{T}_\times^{ij}$ be the dashed-rectangle at i^{th} row and j^{th} column of the reduced $\mathbf{T}_\times$ (right-hand side of Figure 2). $\mathbf{T}_\times^{ij}$ represents the transition probabilities $\mathbf{T}_\times((u,v),(a,b))$ where

$u, v \in V_i$ and $a, b \in V_j$. Let d_v be the maximum degree of nodes in all views (note that $d_v < d$), the number of non-zero elements in a diagonal block $\mathbf{T}_{\times}^{ii}$ is less than $(d_v N)^2$. Since the number of inter-view links between two different views, say views i and j is less than N, the number of non-zero elements in $\mathbf{T}_{\times}^{ij}$ is less than N^2. As a result, the number of non-zero elements in the reduced $\mathbf{T}_{\times}$ is less than $Md_v^2N^2 + (M-1)\,MN^2 \ll (dMN)^2$ when d and M are large. The resulting $\mathbf{K}$ is of the form:

$$\mathbf{K} = \left(\cdots, K_i\left(x, z\right), \cdots\right)^{\mathrm{T}} \tag{8}$$

where $K_i\left(x, z\right)$ is the kernel value of a pair $x, z \in V_i$. Likewise, r_1 has the following form.

$$\mathbf{r}_1 = \left(\cdots r_1^i\left(x, z\right), \cdots\right)^{\mathrm{T}} \tag{9}$$

The preceding steps set the stage for employing efficient numerical algorithms to solve Equation 7 with the reduced forms of $\mathbf{K}$, $\mathbf{T}_{\times}$, and $\mathbf{r}_1$. In our implementation, use the following matrix form of Equation 6 on each individual view to compute the kernel value between two nodes within a view,

$$\mathbf{R}_L^i = \mathbf{r}_1^i + \sum_{j=1}^{M} \mathbf{T}_{\times}^{ij} \mathbf{R}_{L-1}^j \tag{10}$$

where $\mathbf{R}_L^i = \left(\cdots, R_L\left(x, z\right), \cdots\right)^{\mathrm{T}} \in \mathbb{R}^{N^2} \wedge x, z \in V_i$ and $\mathbf{r}_1^i = \left(\cdots, r_1\left(x, z\right), \cdots\right)^{\mathrm{T}} \in \mathbb{R}^{N^2} \wedge x, z \in V_i$. We iteratively compute R_L^i using Equation 10 until convergence to obtain the kernel value of any pair of nodes within view i (see Appendix for a proof of convergence of kernel calculation using Equation 10). Since each actor node in a view has at most one inter-view link to its representation in any other view, the time complexity of naive matrix multiplication for obtaining R_L^i is $O\left(MLN^3\right)$. The computation of the kernel matrix R_L^i depends on R_{L-1}^j, $j \neq i$. In our current implementation, we employ multiple threads to compute all R_{L-1}^j in parallel, yielding substantial speedups relative to the naive matrix multiplication algorithm. Additional speedup can be obtained by exploiting the sparsity of the matrix.

III. Experiments and Results

IRWK generates one kernel for each view by combining information not only from random walks within each view but also the random walks that traverse from each view, the other views. In this section, we empirically compare the performance of IRWK-based classifiers with several baseline as well as the state-of-the-art methods for labeling actors in multi-view networks using two real-world data sets. We also investigate the sensitivity of IRWK-based classifiers with respect to the choice of the relevant parameters as well as its speed of convergence.

A. Data sets

We crawled two real-world multiple-view social networks: (i) **Last.fm Data Set**. We manually identified 11 disjoint groups (categories) of closely similar size of number of users in the Last.fm music network, then we crawled actors (users) as well as item objects (e.g., tag, artist, track) and relation information among objects and actors in the network. We then generated 12 views on Last.fm actors: ArtistView (2118 actors, 149495 links), EventView (7240 actors, 177000 links), NeighborView (5320 actors, 8387 links), ShoutView (7488 actors, 14486 links), ReleaseView (4132 actors, 129167 links), TagView (1024 actors, 118770 links), TopAlbumView (4122 actors, 128865 links), TopArtistView (6436 actors, 124731 links), TopTagView (1296 actors, 136104 links), TopTrack-View (6164 actors, 87491 links), TrackView (2680 actors, 93358 links), and UserView (10197 actors, 38743 links). Except UserView and NeighborView which are explicit views in Last.fm, the other views (e.g., EventView, TrackView) were constructed based on the number of items shared between two actors. That is, two users were connected if they shared more than a specified number of items. For Last.fm data set, we set the threshold for the number of shared items to be 10. Each group in Last.fm refers to a set of actors who have common interests (e.g., http://www.last.fm/group/Metal denotes a group of actors who are interested in Metal music). (ii) **Flickr Data Set**. We manually identified 10 disjoint groups of consisting of approximately the same number of users, and crawled actors as well as items (e.g., photo, tag) and relations among items, actors, and actors and items from the Flickr photo sharing network. We generated 5 views on Flickr users: CommentView (2358 actors, 13789 links), FavoriteView (2724 actors, 30757 links), PhotoView (4061 actors, 91329 links), TagView (1341 actors, 154620 links), and UserView (6163 actors, 88052 links). Except the UserView which was based on the original connection information from the Flickr, other views were constructed based on the number of shared items between two users. For Flickr, we set the threshold for shared items to 5. Each group in Flickr refers to a community of users who share an interest in similar pictures (e.g., http://www.flickr.com/groups/iowa/ denotes a group of users who are interested in pictures of places or events or individuals associated with state of Iowa). In both data sets, we use the group memberships of actors as class labels to train and test predictive models for labeling actors. Table II shows the summary of the two data sets.

TABLE II: Summary of Data sets

	Last.fm	Flickr
Total number of users	10,197	6,163
Number of views	12	5
Number of groups	11	10
Min number of users in a view	1,024	1,341
Max number of users in a view	10,197	6,163
Min number of links in a view	14,486	13,789
Max number of links in a view	177,000	154,620
Total links of all views	1,325,367	378,547

B. Baseline methods

We compared the IRWK induced feature representation with several representative baseline as well as state-of-the-art alternatives using an one-versus-rest sparse logistic regression

TABLE III: Classification performance of compared methods, respectively on five views of Flickr (views' names were truncated for short). Bold value indicates the best classifier based on paired t-test ($p < 0.01$).

	Accuracy(%)					Macro-F1(%)				
Classifier	Comment	Favorite	Photo	Tag	User	Comment	Favorite	Photo	Tag	User
EdgeCluster	44.48	42.92	39.01	38.09	43.14	29.76	30.32	35.61	21.11	45.52
DeepWalk	45.33	47.94	43.81	37.14	50.45	25.74	30.19	37.96	11.69	46.32
LINE(1st + 2nd)	45.51	48.12	43.66	**42.06**	50.49	23.07	29.36	39.15	18.13	**47.32**
Node2Vec	43.81	46.95	43.49	35.87	48.56	24.51	32.53	38.42	16.12	44.43
RWK	**50.86**	50.19	44.41	31.62	50.09	**33.87**	**34.83**	40.46	13.28	46.63
IRWK	**50.76**	**51.73**	**46.81**	39.08	**51.52**	31.47	**35.12**	**42.96**	**21.73**	**47.11**

TABLE IV: Classification performance of compared methods, respectively on twelve views of Last.fm (views' names were truncated for short). Bold value indicates the best classifier based on paired t-test ($p < 0.01$).

		Artist	Event	Nbor	Release	Shout	Tag	TopAlb	TopArt	TopTag	TopTrk	Track	User
Accuracy(%)	EdgeCluster	37.38	33.82	40.29	30.67	47.25	26.58	30.94	53.05	**31.49**	38.63	48.72	52.51
	DeepWalk	36.41	35.08	42.69	36.57	48.02	13.67	37.75	57.74	20.14	43.11	50.86	59.08
	LINE(1st+2nd)	**40.98**	37.79	32.27	37.34	43.36	26.47	36.92	59.21	30.41	43.36	53.55	57.11
	Node2Vec	36.12	35.93	41.64	35.79	46.51	16.99	34.98	56.84	22.45	43.88	50.82	56.25
	RWK	37.87	46.81	47.86	39.21	53.12	**40.42**	39.25	60.95	28.78	50.24	**62.09**	60.91
	IRWK	39.71	**51.85**	**52.91**	**41.94**	**60.32**	**40.43**	**41.71**	**61.93**	28.78	**53.42**	**61.53**	**63.21**
Macro-F1(%)	EdgeCluster	30.13	29.28	38.25	27.11	46.02	**20.29**	27.01	49.78	**23.71**	36.01	38.48	51.04
	DeepWalk	25.51	27.06	37.48	27.59	44.99	6.67	28.25	53.88	6.95	38.32	39.21	56.98
	LINE(1st+2nd)	32.02	31.37	29.32	30.98	41.54	18.51	29.98	55.42	19.77	39.65	42.26	55.64
	Node2Vec	27.93	28.75	37.26	29.18	43.68	12.17	28.15	53.11	13.61	39.08	39.36	54.11
	RWK	31.86	28.65	41.19	30.38	47.93	9.59	29.84	55.36	4.97	44.43	**44.61**	56.07
	IRWK	**35.85**	**36.12**	**45.65**	**33.18**	**54.65**	9.61	**32.48**	**56.61**	4.97	**47.71**	42.82	**58.38**

classifier (trained using the LibLinear package[5]) for labeling actors in multi-view social networks:

- Node2Vec [31]: A method that learns a mapping of nodes in networks to a low-dimensional space of features so as to maximize the likelihood of preserving network neighborhoods of nodes, that has been shown to outperform other methods on the node labeling and link prediction tasks on several real-world networks from diverse domains.
- LINE [30]: A scalable method for local as well as global structure preserving embedding of large (directed or undirected, weighted or unweighted) networks into low-dimensional vector spaces for use in visualization, node labeling, and link prediction, which has been shown effective on a variety of real-world networks.
- DeepWalk [11]: A scalable method for learning latent representations of nodes in a network that encode social relations in a continuous vector space, by constructing language models over sequences of nodes obtained using truncated random walks over the network, which has been shown to be effective at node labeling and related tasks in a variety of social networks.
- EdgeCluster [1]: A scalable method that extracts a sparse representation of an actor's social dimensions (actor's latent affiliations) using an edge-centric clustering scheme .
- RWK[17]: A random walk graph kernel that does not take into account the information from traversal across views.

For the baseline methods, we explored a range of parameter settings on both data sets. We report results using parameter settings that gave the best results on both data sets. Specifically, we explored the performance of Node2Vec, LINE, DeepWalk with the number of dimensions set to: 64, 128, 256, 512. For Node2Vec, for each setting of the number of dimension, we explored several choices for the number of epochs of stochastic gradient descent (SGD): 10, 20, 30, 40, 50, 60, and 70. We settled on the number of dimensions = 128 and the number of SGD epochs = 50 for Node2Vec and the number of dimensions = 256 for both DeepWalk and LINE. LINE offers two kinds of proximities: first-order proximity (proximity based on local structure of network) and second-order proximity (proximity based on the global structure of the network). We explored representations generated using three variants of LINE: first-order LINE, i.e., LINE(1st), second-order LINE, i.e., LINE(2nd) and the concatenation of the two, i.e., LINE(1st+2nd). We report the experimental results on LINE(1st+2nd) since it yields the best results among the three variants of LINE. For EdgeCluster, we explored the following settings for the number of dimensions: 500, 1000, 2000, 3000, 4000, 5000, and 6000; and observed that the best performance was achieved using the number of dimensions = 4000. All other user-specified parameters of EdgeCluster, DeepWalk, LINE, and Node2Vec, were set to their default values. For IRWK, we set the stopping probability $\rho = 0.1$ and the *inter-view* transition probability be equal to 0.05 for all nodes in the multi-view network. Likewise, we also set the stopping probability $\rho = 0.1$ for RWK. For both IRWK and RWK, we normalize kernel matrix for each view, i.e., $K_i(x,z) = K_i(x,z) / \sqrt{K_i(x,x) K_i(z,z)}$ (Note that kernel value is independent of the starting probability τ).

[5]https://www.csie.ntu.edu.tw/~cjlin/liblinear/

TABLE V: Classification performance of compared methods on multi-view setting for Flickr data set. Bold value indicates the best classifier based on paired t-test ($p < 0.05$).

	Classifier	10%	20%	30%	40%	50%	60%	70%	80%	90%
Accuracy(%)	EdgeCluster	39.86	42.04	43.21	43.32	43.57	44.21	44.04	44.56	44.48
	EdgeCluster+	41.61	44.06	45.29	45.73	45.93	46.02	46.45	46.69	46.37
	DeepWalk	46.49	47.64	48.52	48.49	49.63	49.48	49.83	50.24	50.63
	DeepWalk+	47.17	51.52	52.93	54.07	54.64	**55.35**	**55.49**	**55.27**	55.79
	LINE(1st+2nd)	40.13	45.99	48.37	49.88	51.19	52.26	52.74	53.26	52.94
	LINE(1st+2nd)+	43.34	49.95	52.61	54.09	**55.11**	**55.26**	55.35	**55.63**	**56.21**
	Node2Vec	41.09	44.51	46.33	47.51	48.11	48.68	49.11	48.92	49.23
	Node2Vec+	41.98	46.67	49.34	50.88	51.91	52.26	52.56	52.87	53.51
	RWK	43.21	46.23	47.22	47.95	48.27	48.81	48.93	49.16	49.18
	RWK+	46.98	50.86	52.42	52.96	53.95	54.95	55.28	55.19	55.46
	IRWK	46.39	48.44	49.38	50.34	50.89	51.14	51.36	51.46	50.91
	IRWK+	**48.99**	**52.93**	**54.44**	**55.08**	**55.43**	**55.54**	**55.86**	**55.62**	55.73
Macro-F1(%)	EdgeCluster	37.18	39.63	41.01	41.29	41.59	42.13	42.18	42.75	42.75
	EdgeCluster+	38.36	41.56	43.27	43.81	44.02	44.36	44.98	45.26	44.99
	DeepWalk	41.17	42.95	44.02	44.02	45.58	45.25	45.89	46.21	46.77
	DeepWalk+	43.73	49.17	50.89	52.21	52.91	53.83	53.96	53.82	54.37
	LINE(1st+2nd)	31.46	40.81	44.27	46.57	48.05	49.36	49.86	50.43	50.32
	LINE(1st+2nd)+	39.75	48.28	**51.38**	**53.15**	**54.31**	**54.52**	**54.84**	**55.06**	**55.69**
	Node2Vec	37.49	41.01	43.09	44.51	45.19	45.51	46.15	45.82	45.76
	Node2Vec+	39.01	44.47	47.53	49.28	50.49	50.94	51.34	51.62	52.57
	RWK	34.45	39.16	41.01	42.44	43.02	43.98	44.34	44.71	44.69
	RWK+	42.13	47.25	49.77	50.56	52.11	52.49	53.68	53.77	54.11
	IRWK	39.44	42.35	43.67	44.96	45.82	46.43	46.51	46.77	46.11
	IRWK+	**44.52**	**49.82**	**51.74**	**52.81**	53.41	53.66	54.17	54.06	54.18

TABLE VI: Classification performance of compared methods on multi-view setting for Last.fm data set. Bold value indicates the best classifier based on paired t-test ($p < 0.05$).

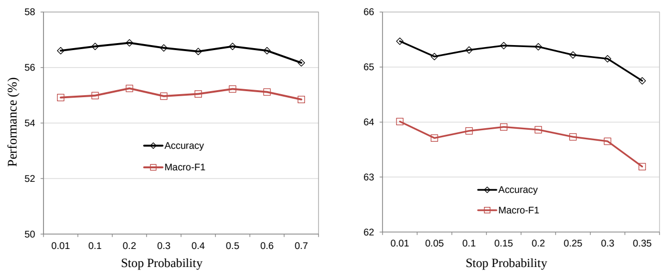
	Classifier	10%	20%	30%	40%	50%	60%	70%	80%	90%
Accuracy(%)	EdgeCluster	48.57	51.12	51.78	52.96	53.37	53.61	53.75	54.12	55.41
	EdgeCluster+	46.36	50.31	52.36	53.19	54.14	54.27	55.01	55.36	55.12
	DeepWalk	51.65	53.24	53.87	54.09	54.36	54.49	54.83	54.84	54.63
	DeepWalk+	**58.26**	**61.17**	**62.31**	**62.87**	**63.41**	63.56	63.86	64.02	64.33
	LINE(1st+2nd)	51.29	55.52	57.39	58.37	59.16	59.61	60.19	60.33	60.37
	LINE(1st+2nd)+	53.31	58.87	60.53	61.55	62.19	62.51	62.87	63.16	63.88
	Node2Vec	52.33	54.38	55.51	56.13	56.57	56.66	57.04	57.03	56.96
	Node2Vec+	54.68	58.66	60.15	61.11	61.66	61.97	62.29	62.45	63.34
	RWK	51.09	53.33	54.52	55.36	55.78	56.05	56.28	56.34	56.36
	RWK+	55.16	59.04	60.24	60.95	61.63	62.03	62.33	62.48	63.19
	IRWK	56.35	59.23	60.61	61.59	62.39	62.58	63.21	62.76	63.06
	IRWK+	56.25	60.19	**61.89**	**62.79**	**63.61**	**64.23**	**64.47**	**64.91**	**65.71**
Macro-F1(%)	EdgeCluster	46.81	49.29	50.16	51.43	51.88	52.15	52.33	52.68	53.94
	EdgeCluster+	44.67	48.66	50.76	51.68	52.66	52.84	53.59	53.92	53.79
	DeepWalk	48.43	50.37	51.21	51.48	51.75	51.83	52.21	52.12	51.88
	DeepWalk+	**55.69**	**59.11**	**60.43**	**61.07**	**61.73**	61.92	62.25	62.48	62.78
	LINE(1st+2nd)	48.92	53.43	55.44	56.57	57.39	57.88	58.52	58.72	58.76
	LINE(1st+2nd)+	50.93	57.03	58.92	60.01	60.71	61.08	61.51	61.77	62.52
	Node2Vec	49.38	51.71	53.03	53.75	54.23	54.33	54.71	54.72	54.54
	Node2Vec+	52.06	56.51	58.16	59.24	59.87	60.23	60.59	60.85	61.67
	RWK	47.59	50.51	52.05	53.13	53.74	54.08	54.37	54.49	54.51
	RWK+	52.02	56.72	58.49	59.31	60.59	60.51	60.86	61.05	61.75
	IRWK	47.48	52.07	54.61	56.02	57.23	57.64	58.33	58.03	58.32
	IRWK+	53.13	57.88	**59.78**	**60.86**	**61.85**	**62.57**	**62.85**	**63.34**	**64.08**

C. IRWK Compared to Baseline Methods

We compare the performance of one-versus-rest sparse logistic regression classifier [33] (as implemented in the LibLinear package[6]) trained to label actors in multi-view social networks using representations produced by IRWK, RWK, Node2Vec, DeepWalk, LINE, EdgeCluster, as described above, using two sets of experiments.

[6]https://www.csie.ntu.edu.tw/~cjlin/liblinear/

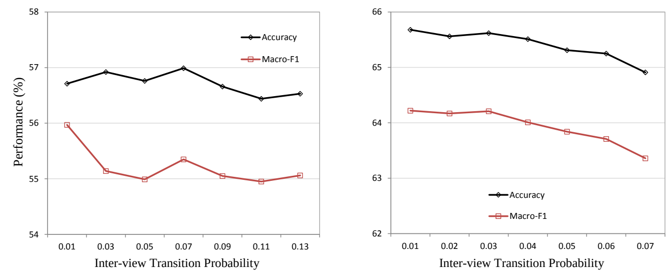
1) Labeling Actors Based on Single Views: The first set of experiments investigates the benefits of IRWK, which, unlike other methods, incorporates information from other views (via inter-view links), even when the actors in multi-view networks are labeled based primarily on information provided by a single view. The performance of the methods as measured by accuracy and Macro-F1, averaged over 10 runs of a 10-fold cross-validation experiment, on the Flickr and Last.fm data sets, are shown in Tables III and IV respectively. It is evident that IRWK, the only method that incorporates



(a) Performance of IRWK as a Function of Stop Probability on Flickr

(b) Performance of IRWK as a Function of Stop Probability on Last.fm

Fig. 3: Sensitivity of IRWK w.r.t Stop Probability



(a) Performance of IRWK as a Function of Inter-view Probability on Flickr

(b) Performance of IRWK as a Function of Inter-view Probability on Last.fm

Fig. 4: Sensitivity of IRWK w.r.t Inter-view Transition Probability

information from other views (through inter-view random walks) in computing the similarity between nodes in each view, significantly ($p < 0.01$) outperforms, in almost all cases, all other representations (RWK, Node2Vec, DeepWalk, LINE, EdgeCluster) that rely solely on information available within the view. RWK comes in at a close second. Node2Vec and DeepWalk show comparable performance, perhaps explained by the fact that both use the Skip-gram model [34] to learn vector space representations of nodes.

2) Labeling Actors Based on all Views: The second set of experiments investigates the benefits of integrating information from all of the views in labeling actors in multi-view networks. To ensure fair comparison between IRWK, a method that is specifically designed to integrate information from within as well as across views and some of the other methods which are not, we compare two variants of each of the methods: (i) We assemble a single integrated network that includes all of the links of each actor from all of the views in which the actor appears. The resulting networks contain 274,967 and 886,894 links, in the case of Flickr and Last.fm, respectively. Since it is not straightforward to use the joined network in the case of IRWK, we choose to apply IRWK on the view that has the largest number of actors and use it as the first variant of IRWK. (ii) In the case of each method, we extract for each actor, a feature vector from each view. We then concatenate the resulting feature vectors into a single feature vector. We add a "+" to the name of the method to denote this variant. For example, the classifier that uses a concatenation of feature vectors obtained by applying EdgeCluster to the individual views is denoted by EdgeCluster+. We compare the performance of the different methods as a function of the percentage of actors in the network with known labels. For each choice of the percentage of labeled actors, we randomly select the corresponding fraction of labeled data for each node label for training and the rest for testing. We repeat this process 10 times and report the average accuracy and Macro-F1.

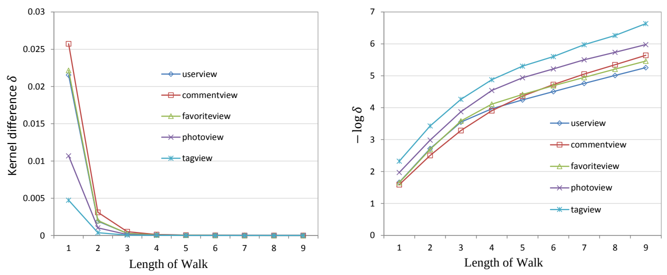
The results of this set of experiments, on Flickr and Last.fm data sets, respectively, are summarized in Tables V and VI. On the Flickr data, IRWK+ outperform all other methods in almost all cases. DeepWalk+ comes in at a close second, with LINE(1st+2nd) catching up both when the fraction of actors with known labels exceeds 50%. On the Last.fm data, IRWK+ and DeepWalk+ are competitive with each other and outperform the other methods with LINE(1st+2nd)+ catching up with both when the fraction of actors with known labels exceeds 50%. The superior performance of IRWK+ is especially impressive in light of the fact that the user-specified parameters of the other methods were set empirically to the values that result in their best performance. It is worth noting that although the dimensionality of the feature space of IRWK is equal to the number of actors in the network, the sparse logistic regression classifier trained on the IRWK-induced representation results in a model that uses only 7% and 3% of the features, respectively on the Flickr and Last.fm data, yielding classifiers that are comparable in their complexity with those obtained using the other methods (data not shown due to space constraints).

We further note that the performance of the two variants of IRWK, RWK, DeepWalk, LINE(1st+2nd), and Node2Vec are, in several cases, comparable, which is perhaps explained by the fact that IRWK, DeepWalk, Node2Vec are, at their core, random walk-based methods. LINE(1st+2nd) is the combination of LINE(1st) and LINE(2nd). LINE(1st) encodes the local structure of the network, similar in a manner that is similar to that done by the first iteration of IRWK in Equation 3. LINE(2nd) encodes the global structure of the network, not unlike that done by random walks of IRWK in Equation 2.

D. Parameter Sensitivity

We investigate how the performance of IRWK varies as a function of the values of stopping probability and the inter-view transition probability. In each case, we report the results of accuracy and Macro-F1 averaged over the runs of a 10-fold cross-validation experiment.

To examine the sensitivity of the performance of IRWK with respect to the stopping probability, we fix the inter-view transition probability to 0.05. We increase the value of stopping probability, i.e., ρ, from 0.01, in steps of 0.01 such that $1 - \rho - 0.05 \times (M - 1) > 0$. Figure 3 shows that the performance of IRWK is fairly stable across a broad range of settings of stopping probability on both the data sets (i.e., the variation in accuracy and Macro-F1 of IRWK as a function of



(a) Kernel Convergence as a Function of Walk Length

(b) Negative Log Kernel Convergence as a Function of Walk Length

Fig. 5: Kernel Convergence w.r.t Length of Walk on Flickr

stopping probability remains approximately under 1% on both Flickr and Last.fm).

To examine the sensitivity of the performance of IRWK with respect to the inter-view transition probability, we fix the stopping probability to 0.1. We increase the value of the inter-view transition probability from 0.01, in steps of 0.01 such that $0.9-(M-1)\times P_{i\to j}>0$. Figure 4 shows the performance of IRWK as a function of inter-view transition probability. Again, we see that the performance of IRWK is stable across a broad range of settings of the inter-view transition probability.

E. Convergence

We investigate the convergence of IRWK as a function of the walk length. Specifically, we define the kernel difference function $\delta(L)=\|R_L-R_{L-1}\|_F / \|R_{L-1}\|_F$, $L>1$, ($\|\cdot\|_F$ denotes for the Frobenius norm a matrix) as a function of length of walk (see Equation 10). Figure 5a shows the value of $\delta(L)$ as a function of the walk length over five views of the Flickr data set. Figure 5b shows the negative of log of $\delta(L)$ as a function of the walk length. The results show that the random walk converges rapidly after a few steps of walk. For example, in Flickr $\delta(L)<0.00005$ when the length of walk is equal or greater than 5 on all views.

IV. Summary and Discussion

Real world social networks typically consist of actors (individuals) that are linked to other actors or different types of objects via links of multiple types. Different types of links relationships induce different *views* of the underlying social network. In this paper, we considered the problem of labeling actors in such multi-view networks: Given a social network in which only some of the actors are labeled, our goal is to label the rest of the actors. We introduced a new inter-graph random walk kernel, namely IRWK, for multi-view network i.e, network with multiple types of links. We used IRWK to generate kernels where each of them is for one individual view. We use the resulting IRWK kernels to train classifiers for labeling actors in a multi-view social network. The results of our experiments on two real-world multi-view social networks show that: (i) IRWK classifiers outperform or are competitive with several state-of-the-art methods for labeling actors in a social network; (ii) IRWKs are robust with respect to different choices of user-specified parameters; and (iii) IRWK kernel computation converges very fast within a few iterations.

Some promising directions for further research include investigation of: (i) more sophisticated models of multi-view networks that exploit the dependencies between the intra-view and inter-view contexts of actors as well as the dependencies among the views, e.g., by taking advantage of latent space generative models; (ii) variants of IRWK that take into account not only the topological similarity of the graphs, but also their semantic similarity, by incorporating node and edge types in kernel calculations; (iii) variants of IRWK that make use of abstractions defined over node and link types; (iv) more extensive empirical study of the algorithms for labeling actors in multi-view networks on a broader range of real-world networks; (v) variants of IRWK and related methods for link prediction in multi-view networks; (vi) variants of methods for node labeling and link prediction, respectively, to node regression and link regression where the labels on the nodes and link assume real values as opposed to categorical values.

Appendix

Derivations for Equations 6, 7, 10, and convergence of 10. From $R_L(x,z)=\sum_{l=1}^{L} r_l(x,z)$, we have

$$
\begin{aligned}
R_\infty(x,z) &= \lim_{L\to\infty}\sum_{l=1}^{L} r_l(x,z)\\
&= r_1(x,z)+\lim_{L\to\infty}\sum_{l=2}^{L} r_l(x,z)\\
&= r_1(x,z)+\lim_{L\to\infty}\sum_{l=2}^{L}\sum_{x_1,z_1} P_t(x_1|x)\,R_0(x_1,z_1)\\
&\times P_t(z_1|z)\,r_{l-1}(x_1,z_1)\\
&= r_1(x,z)+\sum_{x_1,z_1}\mathbf{T}_\times((x,z),(x_1,z_1))\lim_{L\to\infty}\sum_{l=1}^{L} r_l(x_1,z_1)\\
&= r_1(x,z)+\sum_{x_1,z_1}\mathbf{T}_\times((x,z),(x_1,z_1))\,R_\infty(x_1,z_1)
\end{aligned}
$$

We can write Equation 6 in matrix form as follows,

$$
\mathbf{R}_\infty=\mathbf{r}_1+\mathbf{T}_\times\mathbf{R}_\infty \Leftrightarrow \mathbf{R}_\infty=(\mathbf{I}-\mathbf{T}_\times)^{-1}\mathbf{r}_1
$$

where $\mathbf{R}_\infty=(\cdots,R_\infty(x,z),\cdots)^{\mathrm{T}}\in\mathbb{R}^{M^2N^2}$. Replacing $\mathbf{R}_\infty$ in Equation 5, we obtain Equation 7. Hence, we have:

$$
\begin{aligned}
R_L^i(x,z) &= r_1^i(x,z)+\sum_{x_1,z_1}\mathbf{T}_\times\left((x,z)_{ii},(x_1,z_1)\right)\\
&\times R_{L-1}(x_1,z_1) \qquad (11)
\end{aligned}
$$

where $R_L^i(x,z)$ denotes the $R_L(x,z)$ of $x,z \in V_i$. From Lemma 1, we have x_1, z_1 are in the same view, say view j. So, Equation 11 is written as follows.

$$R_L^i(x,z) = r_1^i(x,z) + \sum_{j=1}^{M} \sum_{x_1,z_1 \in V_j} \mathbf{T}_\times\left((x,z)_{ii}, (x_1,z_1)_{jj}\right) \times R_{L-1}^j(x_1,z_1) \quad (12)$$

Hence, Equation 10 is the matrix form of Equation 12. From Equation 12, we have:

$$\begin{aligned} &R_L^i(x,z) - R_{L+1}^i(x,z) = \\ &\sum_{j=1}^{M} \sum_{x_1,z_1 \in V_j} \left(\mathbf{T}_\times\left((x,z)_{ii}, (x_1,z_1)_{jj}\right)\right. \\ &\times \left(\sum_{k=1}^{M} \sum_{x_2,y_2 \in V_k} \mathbf{T}_\times\left((x_1,z_1)_{jj}, (x_2,z_2)_{kk}\right)\right. \\ &\times \qquad \cdots\cdots\cdots \\ &\times \left(1 - \sum_{m=1}^{M} \sum_{x_{L+1},z_{L+1}} \mathbf{T}_\times\left((x_L,z_L)_{nn},\right.\right. \\ &\left.\left.\left.\left.\times\, (x_{L+1},z_{L+1})_{mm}\right) P_e(x_{L+1})\, P_e(z_{L+1})\right)\cdots\right)\right) \end{aligned} \quad (13)$$

Since $\sum_{m=1}^{M}\sum_{x_{L+1},z_{L+1}} \mathbf{T}_\times\left((x_L,z_L)_{nn}, (x_{L+1},z_{L+1})_{mm}\right)$ $= \sum_{m=1}^{M}\sum_{x_{L+1},z_{L+1}} P_t(x_{L+1}|x_L) P_t(z_{L+1}|z_L) R_0^m(x_{L+1},z_{L+1})$ $< (1-\rho)^2 < 1$, $R_L^i(x,z) - R_{L+1}^i(x,z) > 0$. Hence, $R_L^i(x,z)$ converges when $L \to \infty$ (ratio test).

Acknowledgments

This research was supported in by the Edward Frymoyer Endowed Chair in Information Sciences and Technology held by Vasant Honavar at the Pennsylvania State University and in part by the National Science Foundation (NSF) through grant SHF 1518732, and by the National Center for Advancing Translational Sciences, National Institutes of Health (NIH), through Grant UL1 TR002014-01. The content is solely the responsibility of the authors and does not necessarily represent the official views of NSF or NIH. This work has benefited from discussions with the members of the Artificial Intelligence Research Laboratory at Pennsylvania State University.

References

[1] L. Tang and H. Liu, "Scalable learning of collective behavior based on sparse social dimensions," in *CIKM*, pp. 1107–1116, 2009.

[2] M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale, and D. Pedreschi, "Multidimensional networks: foundations of structural analysis," *World Wide Web*, vol. 16, no. 5-6, pp. 567–593, 2013.

[3] M. E. Dickison, M. Magnani, and L. Rossi, *Multilayer Social Networks*. Cambridge University Press, 2016.

[4] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of complex networks*, vol. 2, no. 3, pp. 203–271, 2014.

[5] M. Bazzi, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison, "Community detection in temporal multilayer networks, with an application to correlation networks," *Multiscale Modeling & Simulation*, vol. 14, no. 1, pp. 1–41, 2016.

[6] J. Scott, *Social network analysis*. Sage, 2012.

[7] N. Bui and V. Honavar, "Labeling actors in social networks using a heterogeneous graph kernel," in *SBP*, pp. 27–34, 2014.

[8] Q. Lu and L. Getoor, "Link-based classification," in *ICML*, pp. 496–503, 2003.

[9] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data.," *AI Magazine.*, pp. 93–106, 2008.

[10] L. Tang and H. Liu, "Relational learning via latent social dimensions," in *KDD*, pp. 817–826, 2009.

[11] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, pp. 701–710, ACM, 2014.

[12] H. Eldardiry and J. Neville, "Across-model collective ensemble classification.," in *AAAI*, 2011.

[13] N. Bui and V. Honavar, "On the utility of abstraction in labeling actors in social networks," in *ASONAM*, pp. 692–698, ACM, 2013.

[14] S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study," *J. Mach. Learn. Res.*, vol. 8, pp. 935–983, 2007.

[15] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NIPS*, pp. 321–328, 2004.

[16] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, pp. 912–919, 2003.

[17] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," *Journal of Machine Learning Research*, vol. 11, pp. 1201–1242, 2010.

[18] C. Wang, Y. Song, H. Li, M. Zhang, and J. Han, "Text classification with heterogeneous information network kernels," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[19] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *KDD*, pp. 1365–1374, ACM, 2015.

[20] M. Neumann, R. Garnett, C. Bauckhage, and K. Kersting, "Propagation kernels: efficient graph kernels from propagated information," *Machine Learning*, vol. 102, no. 2, pp. 209–245, 2016.

[21] F. Lin and W. W. Cohen, "Semi-supervised classification of network data using very few labels," in *ASONAM*, pp. 192–199, 2010.

[22] R. Angelova, G. Kasneci, and G. Weikum, "Graffiti: graph-based classification in heterogeneous networks," *World Wide Web Journal*, 2011.

[23] X. Wang and G. Sukthankar, "Multi-label relational neighbor classification using social context features," in *KDD*, pp. 464–472, ACM, 2013.

[24] M. Ji, J. Han, and M. Danilevsky, "Ranking-based classification of heterogeneous information networks," in *KDD*, pp. 1298–1306, 2011.

[25] C. Wan, X. Li, B. Kao, X. Yu, Q. Gu, D. Cheung, and J. Han, "Classification with active learning and meta-paths in heterogeneous information networks," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 443–452, ACM, 2015.

[26] M. Salter-Townshend and T. H. McCormick, "Latent space models for multiview network data," tech. rep., Technical report, Technical Report, 2013.

[27] S. Wang, X. Li, Y. Ye, X. Huang, and Y. Li, "Multi-attribute and relational learning via hypergraph regularized generative model," *Neurocomputing*, 2016.

[28] J. Liu, Y. Jiang, Z. Li, Z.-H. Zhou, and H. Lu, "Partially shared latent factor learning with multiview data," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 6, pp. 1233–1246, 2015.

[29] I. Gollini and T. B. Murphy, "Joint modeling of multiple network views," *Journal of Computational and Graphical Statistics*, vol. 25, no. 1, pp. 246–265, 2016.

[30] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, ACM, 2015.

[31] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*, ACM, 2016.

[32] M. Sugiyama and K. Borgwardt, "Halting in random walk kernels," in *Advances in Neural Information Processing Systems*, pp. 1639–1647, 2015.

[33] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, "An improved glmnet for l1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 13, pp. 1999–2030, June 2012.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.