JMLR: Workshop and Conference Proceedings 29:357–372, 2013 ACML 2013

Coinciding Walk Kernels: Parallel Absorbing Random Walks for Learning with Graphs and Few Labels

Marion Neumann MARION.NEUMANN@UNI-BONN.DE
Universtiy of Bonn, Germany

Roman Garnett RGARNETT@UNI-BONN.DE
Universtiy of Bonn, Germany

Kristian Kersting KRISTIAN.KERSTING@CS.TU-DORTMUND.DE
Technical University of Dortmund, Germany

Editor: Cheng Soon Ong and Tu Bao Ho

Abstract

Exploiting autocorrelation for node-label prediction in networked data has led to great success. However, when dealing with sparsely labeled networks, common in present-day tasks, the autocorrelation assumption is difficult to exploit. Taking a step beyond, we propose the coinciding walk kernel (CWK), a novel kernel leveraging label-structure similarity – the idea that nodes with similarly arranged labels in their local neighbourhoods are likely to have the same label – for learning problems on partially labeled graphs. Inspired by the success of random walk based schemes for the construction of graph kernels, CWK is defined in terms of the probability that the labels encountered during parallel random walks coincide. In addition to its intuitive probabilistic interpretation, coinciding walk kernels outperform existing kernel- and walk-based methods on the task of node-label prediction in sparsely labeled graphs with high label-structure similarity. We also show that computing CWKs is faster than many state-of-the-art kernels on graphs. We evaluate CWKs on several real-world networks, including cocitation and coauthor graphs, as well as a graph of interlinked populated places extracted from the DBpedia knowledge base.

Keywords: learning in graphs and networks, kernels on graphs, random walks, label propagation

1. Introduction

The study of structure in networked data has led to great developments and success in graph-based and collective learning (Sen et al., 2008). In this work, we concern ourselves with learning tasks defined on labeled graphs, when only a subset of the nodes' labels are known. The most straightforward problem is using the available labels to predict those on the remaining nodes. The main hypothesis behind most approaches for node-label prediction is that the labels of instances are autocorrelated (Neville and Jensen, 2005). This stems from the *homophily assumption*, that same-labeled nodes are more likely to link to each other.

Hypothesis 1 (Autocorrelation, homophily)

Nodes that are close to one another in the graph are likely to have the same label.

Exploiting this assumption is often profitable, provided that within a small neighbourhood of each unlabeled node, we have a sufficient amount of label evidence to make confident predictions. Due to the enormous size of present-day networks, however, it is common to have only very few labeled nodes, resulting in having too few observations near many unlabeled nodes to effectively apply Hypothesis 1. In turn, classification gets increasingly more difficult. When data are sparsely labeled, we therefore have to do more than exploiting closeby labels to accurately classify unlabeled nodes. Previous work in this direction has introduced latent graphs by adding additional edges (Gallagher et al., 2008; Shi et al., 2011), run multiple random walks with restarts (Lin and Cohen, 2010a), and suggested schemes for active learning (Ji and Han, 2012).

Here we propose an alternative approach. We move beyond the straightforward homophily assumption, to say that not only are nearby nodes likely to have the same label, but also nodes with similar local structure, where we define "structure" to be "the arrangement and connectivity of labels on nearby nodes."

Hypothesis 2 (Label-structure similarity)

Nodes with similarly arranged labels in their local neighbourhoods are likely to have the same label.

Our main contribution is a new kernel, the *coinciding walk kernel* (CWK),[1] that uses short random walks to quantify how similarly the labels surrounding each node are arranged.

Random walks (RWs) in general enjoy huge popularity in graph-based learning and have proven a powerful tool both for defining kernels on graphs (defined between nodes of a graph) and graph kernels (where graphs are themselves inputs to the kernel).[2] A common idea in the graph kernel community is to measure the similarity of two labeled graphs by analyzing the labels encountered during random walks on the respective graphs (Gärtner et al., 2003; Kashima et al., 2003; Neumann et al., 2012); the last reference used this idea to design a kernel among partially labeled graphs. CWKs are inspired by the construction of these graph kernels; however, they define a kernel among the nodes of a graph. Common kernels on graphs include the diffusion kernel (Kondor and Lafferty, 2002), the p-step random walk kernel (Smola and Kondor, 2003), and the Moore–Penrose pseudoinverse of the Laplacian, L^+, (Fouss et al., 2012) which is a limiting case of the regularized Laplacian kernel (Smola and Kondor, 2003). All of these kernels have random-walk interpretations; however, none of them considers known labels during their computation, and as a result, they cannot take advantage of Hypothesis 2.

We view known node labels as providing valuable information that should be considered in the construction of a kernel used for node-label prediction. More precisely, *partially absorbing random walks* (PARWs), where, with some probability, the walks stop progressing once they hit a label, give the known labels influence over the walk process (Zhu et al., 2003; Wu et al., 2012). We consider the distribution over sequences of labels encountered during a PARW from a node as encoding its "label structure." To address Hypothesis 2, we then define the CWK between two nodes to be the probability that parallel PARWs leaving from those nodes coincide, that is, hit the same label at the same time. By lifting the random walk

1. Portions of this work appeared in (Neumann et al., 2013).
2. We make this distinction between "kernels on graphs" and "graph kernels" throughout.

from being on the nodes of a graph to being on its labels, two nodes can be similar even if they are very distant from each other in the graph, or even on disconnected graphs. On the other hand, two PARWs could encounter similar label sequences simply by virtue of having left from nearby nodes in the graph, so the CWK is also compatible with Hypothesis 1.

Most RW-based approaches, absorbing or not, only analyse the walks' steady-state distributions (Kondor and Lafferty, 2002; Zhu et al., 2003; Lin and Cohen, 2010a; Wu et al., 2012). However, RWs using the graph's row-normalized adjacency matrix as the transition matrix converge to a constant steady-state distribution. To address this, the idea of *early stopping* was successfully introduced in power iteration methods for clustering (Lin and Cohen, 2010b) and node-label prediction (Szummer and Jaakkola, 2001). The insight here is that the intermediate distributions obtained by the RWs during the convergence process are extremely interesting. In this paper, we adopt this idea as well and use the entire evolution of labels encountered during partially absorbing RWs up the a given length as representing local structure, rather than only using the limiting distribution. CWKs therefore substantially leverage inference by aggregating label predictions based on different walk lengths.

The distribution of labels encountered during PARWs clearly depends on the locations and labels of previously observed nodes, connecting the CWK to the concept of data-dependent kernels. Data-dependent kernels are widely used in semi-supervised learning (Zhou et al., 2003; Sindhwani et al., 2005), where the kernels are for example constructed from the Laplacian of a graph modeling the data geometry. Approaches like semi-supervised support vector machines (see (Chapelle et al., 2008) for an extensive comparison and review) then try to enforce smoothness of predictions along a manifold defined by the data in feature space, typically by modifying the optimization objective. In this paper, however, we investigate kernel construction leveraging label-structure information in plain graph data where no features on the nodes are given. This means that the CWK is – in contrast to the kernels used in standard semi-supervised learning – a *label-dependent kernel* rather than a data-dependent kernel. However, both approaches could complement each other under the right circumstances. Previous methods using label information to improve the kernel known as label-dependent kernels or kernel-target alignment have proven successful (Zhu et al., 2004; Min et al., 2007). Thus, by exploiting label absorbing random walks, the CWK is a label-dependent kernel on a graph using the label information directly in the kernel construction. This is in contrast to existing approaches which modify an existing kernel to improve alignment on the labeled data.

To summarize, CWKs combine the benefits of kernel methods and inference approaches in networked data. As our extensive experimental results demonstrate, this can considerably improve node-label prediction, especially in sparsely labeled graphs.
The main contribution of this paper is the introduction of the coinciding walk kernel,

- the first label-dependent kernel on graphs leveraging label information directly in the kernel construction, and thus,
- providing a learning method for node-label classification that intertwines inference and kernels on graphs.

We proceed as follows. We start off by defining the main ingredient of coinciding walk kernels, namely partially absorbing random walks. After introducing the coinciding walk kernel and its probabilistic interpretation, we show its positive definiteness. Then, we will

relate CWKs to other existing methods considering label-structure similarity. Before concluding, we present experimental results on several state-of-the-art graph datasets, and discuss parameter sensitivity and computational complexity of the proposed kernel.

2. Parallel and Partially Absorbing Random Walks

As the main ingredient of coinciding walk kernels – the label-structure similarity of nodes in a graph – is modeled by the probability that parallel RWs coincide, we will now review Markov random walks on graphs. Further, we will explain how we examine label-structure similarity via parallel partially label-absorbing random walks.

2.1. Absorbing Random Walks

Consider a graph $G = (V, E)$ with $|V| = n$ vertices and a set of edges E specified by a weighted adjacency matrix $A \in \mathbb{R}^{n \times n}$. For convenience we take $V = \{1, 2, ..., n\}$. A random walk on G is a Markov process $X = \{X_t : t \geq 0\}$ with a given initial state $X_0 = i$. We will also write $X_t^{(i)}$ to indicate the walk began at i. The probability that the walk jumps from i to j, i.e. the transition probability $T_{ij} = P(X_{t+1} = j \mid X_t = i)$, only depends on the current state $X_t = i$. These one-step transition probabilities for all nodes in V can be easily represented by the row-normalized adjacency or *transition matrix* $T = D^{-1}A$, where $D = \text{diag}(\sum_j A_{ij})$.

Let $S \subseteq V$ be a set of nodes. Given T and S, we define an *absorbing random walk* to have the modified transition probabilities $\hat{T}$, defined as

$$\hat{T}_{ij} = \begin{cases} 0 & \text{if } i \in S \text{ and } i \neq j \\ 1 & \text{if } i \in S \text{ and } i = j \\ T_{ij} & \text{else,} \end{cases} \tag{1}$$

Nodes in S are "absorbing" in that a walk never leaves a node in S after it is encountered.

Now, consider a partially labeled graph $G = (V, E, \ell)$ where $V = V_L \cup V_U$ is the union of labeled and unlabeled nodes, respectively, $\ell \colon V \to [k]$ is a label function with known values for the nodes in V_L, and k is the number of available labels. We will describe how we can monitor the distribution of labels encountered during absorbing RWs on G. Let the matrix $P_0 \in \mathbb{R}^{n \times k}$ give the prior label distributions of all nodes in V. If node $i \in V_L$ is observed with label $\ell(i)$, then the ith row in P_0 is the Kronecker delta distribution concentrating at $\ell(i)$, i.e., $(P_0)_i = \delta_{\ell(i)}$. We initialize the label distributions for the unlabeled nodes V_U with some prior, for example a uniform distribution.[3] The ith row of P_0 now gives the probability distribution for the first label encountered, $\ell(X_0^{(i)})$, for an absorbing RW starting at i. Now, it is easy to see by induction that by iterating the map

$$P_{t+1} \leftarrow \hat{T} P_t, \tag{2}$$

$(P_t)_i$ similarly gives the distribution over $\ell(X_t^{(i)})$.

If we define the absorbing states to be the labeled nodes, $S = V_L$, then the *label propagation* algorithm introduced in (Zhu et al., 2003) can be cast in terms of simulating absorbing

3. This prior could also be the output of an external classifier built on available node attributes.

RWs with transition probabilities as given in Eq. (1) until convergence, then assigning the most probable absorbing label to the nodes in V_U. For the rest of this paper we will refer to this "label-absorbing" random walk just as an absorbing random walk.

2.2. Partially Absorbing Random Walks

Recall that our main goal is to define a kernel on a graph to perform learning tasks like node classification in sparsely labeled networks based on autocorrelation and label-structure similarity. Utilizing RWs with fully absorbing states at the labeled nodes as defined above, however, is somewhat restrictive towards this goal – only the first label encountered will have any impact on the evolution of a particular RW. This is compatible with the homophily hypothesis, but not very useful for capturing the structure of surrounding labels. Hence, we have to soften the definition of absorbing states. This can be naturally achieved by employing partially absorbing random walks (PARWs) (Wu et al., 2012).

The simplest way to define PARWs, in the setting of label-absorbing RWs considered here, is to extend our graph G by adding a special node for each label in $[k]$ and adding edges from each labeled node $i \in V_L$ to its respective label node. We then make these auxiliary nodes absorbing states and vary the transition probabilities from the labeled nodes to them. The transition probabilities in this graph $\tilde{G} = (V \cup [k], \tilde{E})$ are given by $\tilde{T}$ having the following block structure:

$$\tilde{T} = \begin{bmatrix} T_{U,U} & T_{U,L} & 0 \\ (1-\alpha)\, T_{L,U} & (1-\alpha)\, T_{L,L} & \alpha\, \delta_L \\ 0 & 0 & I \end{bmatrix}, \tag{3}$$

where $\alpha \in [0, 1]$ is the absorbing probability. Note that by setting $\alpha = 1$ we can exactly model the fully absorbing RWs defined previously. On the other hand, by setting $\alpha = 0$ we get a simple power iteration with constant steady-state distribution. When using the latter setting for learning it is crucial to apply some kind of early termination in order to learn meaningful clusters or class labels (Szummer and Jaakkola, 2001; Lin and Cohen, 2010b). We will utilize PARWs for our coinciding walk kernel on graphs by combining both techniques, partial label propagation and early stopping, into a measure for local structure similarity of the nodes in a graph.

2.3. Parallel Absorbing Random Walks

The final ingredient we need are parallel random walks, as they allow one to refer to the sequences of states of two or more random walks of the same length. Co-occurring RWs can be used to describe the similarity of either entire graphs or nodes in a graph based on the structure of the local neighbourhood of the nodes. These similarities will be the basis of the coinciding walk kernel defined in the next section. Let us now give a formal definition of parallel random walks. A parallel random walk of length $t_{\max}$ among a set of nodes S is given by the sequences $\{X_t^{(i)}\}_{0 \le t \le t_{\max}}$ of $t_{\max}$ states visited by the random walks starting at the respective nodes $i \in S$. Parallel partially absorbing random walks are given by straightforwardly combining the according definitions.

Algorithm 1 CWK computation

Input: max walk length $t_{\max}$, absorbing rate α, inital label distributions $P_0 \in \mathbb{R}^{n\times k}$, adjacency matrix $A \in \mathbb{R}^{n\times n}$
Output: coinciding walk kernel K_{CW}

```
K ← P_0 P_0^⊤
T = D^{-1} A, where D = diag(Σ_j A_ij)
T̃ ← CONSTRUCT-TRANS(T, α)                    ▷ cf. Eq. (3)
for t ← 1...t_max do
    P̃_t ← T̃ P̃_{t-1}                          ▷ one step transition
    P_t ← (P̃_t)_i, i ∈ {1, .., n}
    K ← K + P_t P_t^⊤                        ▷ add kernel contribution
end for
K_CW ← 1/(t_max+1) K                         ▷ normalize kernel
```

3. Coinciding Walk Kernel

Now, we can define the coinciding walk kernel, which is the main contribution of our work. The intuition underlying CWKs is simple: PARWs on partially labeled graphs encode both label and structure similarity. Thus, CWKs can exploit Hypotheses 1 and 2 for learning tasks on graphs. Before we show that K_{CW} is a valid kernel, we discuss its probabilistic interpretation as well as some interesting properties.

3.1. Definition and Random Walk Interpretation

The coinciding random walk kernel on a graph $G = (V, E)$ is defined as

$$K_{\text{CW}} = \frac{1}{t_{\max}+1} \sum_{t=0}^{t_{\max}} P_t P_t^\top, \tag{4}$$

where the matrices of label probabilities $P_t \in \mathbb{R}^{n\times k}$ are obtained by replacing $\hat{T}$ by $\tilde{T}$ in Eq.(2) and considering the respective entries in the extended label probability matrix $\tilde{P}_t$,

$$P_t = (\tilde{P}_t)_{i\in V}, \quad \text{and} \tag{5}$$

$$\tilde{P}_{t+1} \leftarrow \tilde{T}\tilde{P}_t. \tag{6}$$

Note that $\tilde{P}_t \in \mathbb{R}^{n+k\times k}$ is simply the probability matrix P_t extended by a $k \times k$ identity matrix. K_{CW} has two kernel parameters: the absorbing probability α, and the maximum walk length $t_{\max}$, where α controls trade-off between the homophily and label-structure similarity assumptions.

The matrix $(P_t)_i(P_t)_j^\top$ can be interpreted as the probability that parallel PARWs leaving from i and j are on nodes with the same label at time t, that is, that $\ell(X_t^{(i)}) = \ell(X_t^{(j)})$. Hence, CWKs have the following intuitive interpretation: the value of the coinciding walk kernel for two nodes i and j is the probability that parallel PARWs of length $t_{\max}$ starting from i and j encounter the same label at any given time $0 \le t \le t_{\max}$.

Theorem 1

K_{CW} as defined in Eq. (4) is positive-semi definite (i.e., is a valid Mercer kernel).

It is obvious that K_{CW} is a positive-semi definite kernel as it is the scaled sum of polynomial kernels $k(x, y) = (x^\top y + c)^d$, with $c = 0$ and $d = 1$, i.e., $K_{\mathrm{CW}}(i, j) \propto \sum_{t=0}^{t_{\max}} (P_t)_i (P_t)_j^\top$.

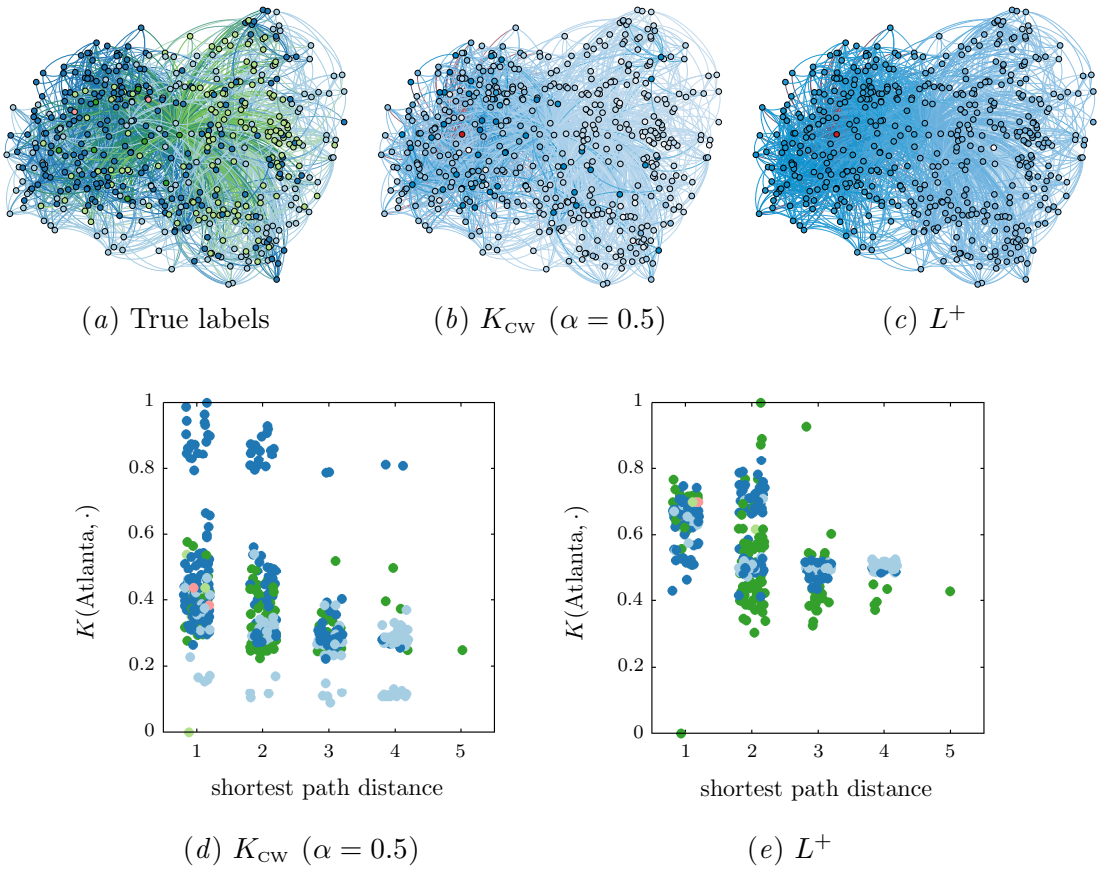
The computation of CWK on a graph G is summarized in Algorithm 1. The computational complexities of the required naïve calculations are $\mathcal{O}(k\, t_{\max}\, |E|\, n)$ for the one step transition and $\mathcal{O}(k\, t_{\max}\, n^2)$ for the kernel contribution, where $|E|$ is the number of edges. It is worth mentioning that for most learning tasks it is sufficient to compute the train–train and train–test fractions of the kernel matrix. This can be accomplished efficiently by precomputing the $\{P_t\}$ and summing only the required outer products with a complexity of $\mathcal{O}(k\, t_{\max}\, |V_L| n)$. Algorithm 1 has an overall computational complexity of $\mathcal{O}(k\, t_{\max}\, |E|\, n)$, however, the kernel computation for sparse graphs (small $|E|$) with few labeled nodes ($|V_L| \ll n$) is efficient.

In Figure 1 we provide an illustration of CWK on a subgraph of a labeled graph built from concepts in the DBpedia ontology marked as "populated places."[4] Each concept is a node in our graph and is backed by a Wikipedia page. We added an undirected edge between two places if one of their corresponding Wikipedia pages links to the other. The DBpedia ontology further divides populated places into "countries," "administrative regions," "cities," "towns," and "villages;" these five labels serve as class labels. This example was chosen because the resulting graph does not necessarily exhibit homophily; for example, villages (approximately half the dataset) are much more likely to link to countries than to other villages. For our illustration, we built a graph with $|V| = 500$ nodes by taking a breadth-first search from "Atlanta." We then calculate the pseudoinverse of the Laplacian kernel (L^+) as well as the coinciding walk kernel (with $\alpha = 0.5$ and $t_{\max} = 10$), using a random selection of 20% of the nodes for V_L. Atlanta was not among the labeled nodes. The rows of K_{CW} corresponding to $K(\text{Atlanta}, \cdot)$ are illustrated in Figure 1 (b) and (d). One can clearly see that CWK is able to capture structure similarity as several distant nodes have high values and nearby nodes including nodes in the direct neighbourhood of Atlanta show low values. The rows of L^+ are shown in Figure 1 (c) and (e). We can see that L^+ (on average) decreases smoothly with increasing distance from Atlanta (reflecting the homophily assumption); whereas the value of K_{CW} also shows some highly correlated far-away nodes, as well as less correlated nearby nodes. Moreover, the magnitude of K_{CW} is highly correlated with the correct label ("city") – the highest kernel values are exclusively achieved by other cities throughout the network, exactly the behavior desired for predicting Atlanta's label. It is also interesting to note that the lowest kernel values are exclusively among nodes in the "town" class, perhaps due to strikingly different label structure in their neighbourhoods.

3.2. Learning with Structure Similarity

Before presenting our experimental results, we will describe one of the baseline approaches and briefly review related work on learning with structure similarity. The closest approach to CWKs, also incorporating local structure similarity, is introduced in (Desrosiers and Karypis,

4. An implementation of the used DBpedia (www.dbpedia.org) graph extractor is available at https://github.com/rmgarnett/dbpedia_graph_extractor.



(a) True labels (b) K_{CW} ($\alpha = 0.5$) (c) L^+

(d) K_{CW} ($\alpha = 0.5$) (e) L^+

Figure 1: **Subgraph of the POPULATED-PLACES Dataset.** Panels (a) - (c) show a subgraph of the POPULATED-PLACES graph extracted from DBpedia consisting of the 500 nearest nodes to the node "Atlanta." The graph layout algorithm used (OpenOrd) was force-directed; nearby nodes have a high connectivity. The edge colours are created by perceptual blending of the colours of the incident nodes. Panel (a) shows the class labels ("country" (green), "administrative region" (light green), "city" (blue), "town" (light blue), "village" (pink)). Panel (b) and (c) illustrate the values of the coinciding walk kernel and L^+ of the kernel row for Atlanta, coloured red. Dark blue means high similarity, i.e., high kernel value, and white represents low similarity. Panels (d) and (e) show scatter plots of the shortest path distance vs. the normalized values of $K(\text{Atlanta}, \cdot)$ for K_{CW} and L^+ respectively, where the colours encode the class labels. Best viewed in colour.

2009). Desrosiers and Karypis use a similarity measure based on parallel RWs with constant termination probability in a relaxation labeling algorithm. We will compare CWK to this method referred to as RL. The similarity measure used in RL corresponds to the probability that parallel random walks with constant termination probability are of exactly the same length and generate exactly the same sequence of labels at all times. Even though RL is slightly more general in the sense of being able to use labeled edges, it has two major drawbacks. First, it has four parameters, α_{RL} and β_{RL}, regulating the influence of label

uncertainty and of the similarity measure in the relaxation labeling iterations, respectively, γ, the constant termination probability, and N, the maximum walk length. Second, the minimal space complexity for node-label prediction scales with the number of unlabeled nodes ($|V_U| \times n$), whereas CWK scales with the number of labeled nodes ($|V_L| \times n$), which is favourable for within-network classification in sparsely labeled graphs. Moreover, their similarity measure $\Sigma^{(t)}$ is used in an iterative relaxation labeling approach in which $\Sigma^{(t)}$ is changing over the iterations, and despite being a kernel[5] it is not clear how to use it directly in a kernel-based approach.

Another approach exploiting the structure of subnetworks is heterogeneous label propagation (Hwang and Kuang, 2010). In contrast to the CWK, heterogeneous LP needs explicitly known subnetworks. Random walks with restart are used as proximity weights for ghost edges in (Gallagher et al., 2008), but then the features considered by a later bag of logistic regression classifiers are only based on a one-step neighbourhood.

4. Experiments

Our intention here is to investigate the power of coinciding walk kernels for the task of node-label prediction in sparsely labeled graphs.[6] We compare their performance to existing methods from the kernel and collective inference community. The main questions to answer are whether CWKs are able to utilize structure similarity and whether employing CWKs improves over state-of-the-art graph-based learning methods on datasets suggesting Hypothesis 2. Additionally, we show that CWKs perform competitive on datasets where mostly homophily (Hypothesis 1) holds. Further, we analyse parameter sensitivity and computational properties of CWKs, i.e., runtime and space complexity of the kernel computation.

4.1. Experimental Protocol

We compare the classification accuracy in several real-world graphs of the following methods:

- CWK: coinciding walk kernels,
- LP: label propagation (Zhu et al., 2003),
- RL: relaxation labeling using structure similarity, (Desrosiers and Karypis, 2009).
- LGC: local and global consistency (Zhou et al., 2003),
- VND: von Neumann diffusion kernel (Zhou et al., 2003; Fouss et al., 2012),
- DIFF: diffusion kernel (Kondor and Lafferty, 2002), and
- L+: pseudoinverse of the (normalized) Laplacian (Fouss et al., 2012).

LP is the obvious baseline approach. RL is currently the most accurate method in the area of collective classification, c.f. results in (Desrosiers and Karypis, 2009). LGC is a diffusion scheme for semi-supervised learning suggested in (Zhou et al., 2003). During the development of their method, Zhou et al. also suggested the kernel $K_{\mathrm{VND}} = (I - \alpha D^{-1/2} A D^{-1/2})^{-1}$

5. Note that Desrosiers and Karypis did not show that their similarity matrix Σ is a kernel, but by taking a slightly different view we can write $\Sigma^{(t)} = \gamma^2 \sum_{j=1}^{N} (1-\gamma^{2j}) \prod_{i=0}^{j} P_i^{(t)} (P_i^{(t)})^\top$, leading to a straightforward proof of positive definiteness for graphs with unlabeled edges.
6. A CWK implementation is available at https://github.com/rmgarnett/coinciding_walk_kernel.

Table 1: Dataset properties. PP-xk is short for POPULATED-PLACES-xk. # graphs indicates the number of connected components and P_{freq} the proportion of the most frequent class. P_{switch} reflects the probability of adjacent nodes switching their labels.

	properties					
dataset	# nodes	# edges	# labels	# graphs	P_{freq}	P_{switch}
PP-1k	1 000	5 253	5	1	43%	69%
PP-3k	3 000	16 546	5	1	50%	66%
PP-5k	5 000	26 648	5	1	53%	70%
WEBKB	1 462	61 766	6	4	28%	33%
DBLP	1 711	2 898	4	1	36%	21%
CORA	2 708	5 278	7	78	30%	18%
CITESEER	3 264	4 536	6	390	21%	26%
PP-100k	100 000	374 480	(used for runtime analysis cf. Fig. 4(b))			

which is the von Neumann diffusion kernel (VND) (Fouss et al., 2012) on the normalized adjacency matrix. Note that, VND is closely related to the regularized Laplacian kernel[7] (Smola and Kondor, 2003). Hence, we choose VND, DIFF and L+ to represent existing successful kernels on graphs. All kernel-based predictions (CWK, VND, DIFF, and L+) are achieved via support vector machine (SVM) classification.

The following graph datasets are used for evaluation:

- POPULATED-PLACES[8] (link graph extracted from DBpedia, described above),
- WEBKB[9] (cocitation graph of webpages from computer science departments of four universities),
- DBLP[10] (connected coauthor graph extracted from the DBLP database),
- CORA[11] (citation network of scientific papers), and
- CITESEER[11] (citation network of scientific papers).

To measure homophily in the datasets, we compute a statistic, P_{switch}, as the probability that a mixed random walk switches labels on adjacent nodes. That is, if P_s is the stationary distribution of the random walk[12] and $P_{\text{switch}|i}$ is the vector of conditional probabilities of switching labels from a given node, then $P_{\text{switch}} = P_s^\top P_{\text{switch}|i}$. Low values of this measure signal the presence of homophily (favouring Hypothesis 1); whereas high values indicate a lack of label smoothness (rejecting Hypothesis 1). For datasets with low P_{switch}, exploiting label-structure similarity (Hypothesis 2) may be more beneficial. P_{switch} and other properties of all datasets are summarized in Table 1. For the POPULATED-PLACES dataset, we created graphs of varying sizes by performing a breadth-first search from the first node in the graph

7. $K_{\text{REGLAP}} = (I - \alpha\tilde{L})^{-1} = (I - \alpha D^{-1/2} L D^{-1/2})^{-1} = (I - \alpha(I - D^{-1/2} A D^{-1/2}))^{-1} = ((1+\alpha)I - \alpha D^{-1/2} A D^{-1/2})^{-1}$, where $\tilde{L}$ is the normalized Laplacian.
8. http://www-kd.iai.uni-bonn.de/pubattachments/727/populated_places.tar.xz
9. http://www.netkit-srl.sourceforge.net/data.html
10. http://www.cs.illinois.edu/homes/mingji1/DBLP_four_area.zip
11. http://www.cs.umd.edu/projects/linqs/projects/lbc/index.html
12. P_s can be calculated as the normalized eigenvector of $T^\top$ with maximal eigenvalue.

Table 2: Average accuracies (%) on 20 test sets of the POPULATED-PLACES datasets, where PP-xk is short for POPULATED-PLACES-xk. Italic indicates statistically significant best performance among the kernel methods and bold indicates statistically significant best performance among all methods both under a paired t-test ($p < 0.05$).

	5%							10%						
	CWK	DIFF	L+	VND	RL	LGC	LP	CWK	DIFF	L+	VND	RL	LGC	LP
PP-1k	***52.4***	44.3	45.2	46.1	42.6	42.8	32.7	***55.0***	50.6	48.5	49.9	46.1	48.2	33.8
PP-3k	61.0	60.2	51.5	52.6	57.7	60.9	44.7	63.3	63.2	53.9	55.8	59.4	62.0	50.2
PP-5k	58.4	59.7	53.4	54.5	53.3	58.8	39.9	***64.0***	61.6	55.5	57.4	59.2	60.4	40.4

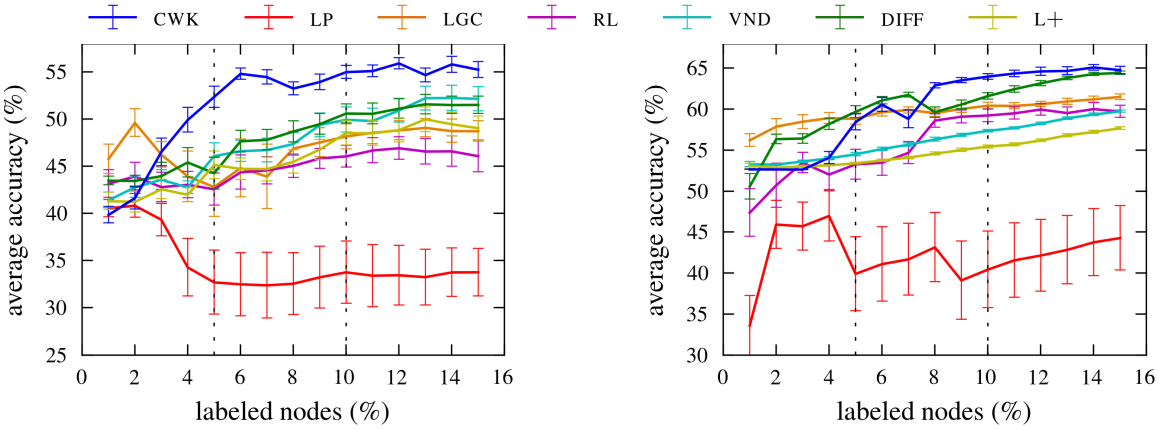
(Alabama). Note that the POPULATED-PLACES datasets have a rather high probability that adjacent nodes are of different labels, i.e. P_{switch} is high. This can also be seen in Figure 1(a). Hence, for these datasets we expect structure similarity to be important for node label prediction. For WEBKB we combined the cocitation networks of all universities (Cornell, Texas, Washington, and Wisconsin) into one disconnected graph.

We focus on sparsely labeled graphs and use 20 randomly generated test splits for 1% up to 15% labeled nodes. The test sets are the same for each method and all reported classification accuracies are an average over the results on the 20 test sets. The performance of all kernel-based classifiers is evaluated by running C-SVM classifications using `libSVM`.[13] Parameter learning is done by the following protocol. For each method we train all parameters (including the SVM cost parameter) jointly via grid search on 10 randomly generated training splits having 5% and 10% labeled nodes. Again, the training sets are the same for each method. For prediction we use the first set of parameters (trained for 5% labeled data) for training percentages from 1% to 7% and the second set of parameters for all scenarios with more than 7% labeled data. The following parameter values were tested: CWK: $t_{\max} \in \{0, 1, \ldots, 10, 20, \ldots, 200\}$, $\alpha \in \{0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 0.8, 0.9, 0.95, 0.98, 0.99, 1\}$; RL: $N \in \{1, 2, \ldots, 5\}$, $\gamma \in \{0.1, 0.3, 0.5, 0.7\}$, $\alpha_{\text{RL}} \in \{0.25, 0.5, \ldots, 1.5\}$, $\beta_{\text{RL}} \in \{0.5, 1.0, \ldots, 3.0\}$; LGC and VND: $\alpha \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 0.8, 0.9, 0.95, 0.98, 0.99\}$; DIFF: $\beta \in 2^{\{-7,\ldots,7\}}$; and all kernel methods: SVM cost $C \in 2^{\{-7,\ldots,7\}}$.

4.2. Predictive Performance

Exploiting Structure Similarity. Here we analyse the performance of all methods on the POPULATED-PLACES datasets where – indicated by a high switching probability P_{switch}, cf. Table 1 – one can expect structure similarity to be useful for classification. The predictive performances for three variations with 1 000, 3 000, and 5 000 nodes (PP-1k, PP-3k, PP-5k) for 5% and 10% labeled nodes are summarized in Table 2. CWK performed significantly better (under a paired t-test with $p < 0.05$) than the comparing methods on three out of six experiments. Only in one of six cases (5% on PP-5k) did DIFF and LGC perform slightly better than CWK; however, the difference was not significant. Hence, exploiting label-structure similarity via partially label absorbing random walks clearly improves over existing graph-based learning on non-homophilic datasets. Figure 2 shows results for 1%

13. http://www.csie.ntu.edu.tw/~cjlin/libsvm/



(*a*) PP-1k (*b*) PP-5k

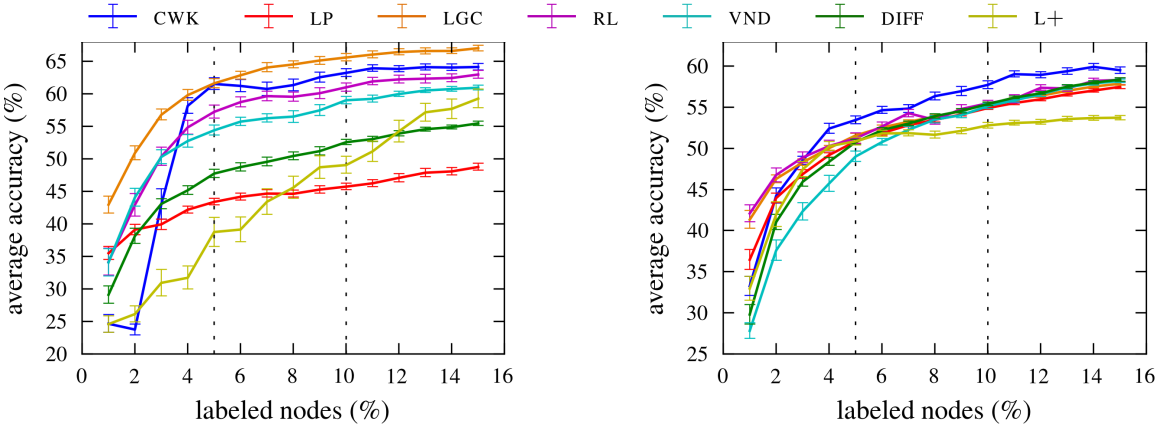
Figure 2: **Average accuracies (and standard errors) on PP-1k and PP-5k.** The dotted lines indicate 5% and 10% training data, corresponding to the results in Table 2. Best viewed in colour.

to 15% labeled nodes for PP-1k and PP-5k. In general, we observe that CWK achieves the best results followed by all other kernels on graphs (VND, DIFF, and L+), LGC and RL. LP fails to accurately predict the labels in the populated places graphs as it relies purely on the homophily assumption and therefore cannot leverage the structure similarity inherent to these networks. Surprisingly, RL does not achieve convincing results either.

Common Benchmark Graphs. The predictive performances for four common benchmark graphs (DBLP, WEBKB, CORA, and CITESEER) with 5% and 10% labeled nodes are summarized in Table 3. These datasets mostly obey the autocorrelation assumption (Hypothesis 1) which can be seen from the rather low label switching probabilities reported in Table 1. In the scenario with 5% labeled nodes, CWK performed significantly better (under a paired t-test with $p < 0.05$) than the comparing kernel methods (DIFF, L+, and VND) on all four datasets. Further, CWK performs significantly best for CITESEER, whereas LGC outperforms all other methods for DBLP and CORA. When considering 10% labeled nodes, CWK performed significantly best among the kernels on graphs in three out of four cases. Comparing all methods, CWK and LCG both win significantly on one dataset. Overall, CWK outperforms all kernels on graphs on a representative sample of common benchmark graphs. These results indicate that incorporating label information into the kernel construction, i.e. using a label-dependent kernel such as CWK for node classification, improves performance over kernels on graphs using graph structure only. Further, CWK perform competitively compared to a range of successful prediction methods from the areas of semi-supervised learning and collective inference. Figure 3 shows average accuracies and standard errors of all compared methods for 1% up to 15% labeled nodes for WEBKB and CITESEER. On WEBKB, CWK is clearly the best performing kernel on graphs and in comparison to all methods it is the second best classifier. On CITESEER, CWK performs significantly better than all baselines for label fractions larger then 3%. As P_{switch} (26%) is fairly low on this dataset, this success might be explained by the huge number of connected components (390) – CWK is able to capture similarities between two nodes in disconnected components; whereas other kernels on graphs always give a value of zero in these cases.

Table 3: Average accuracies (%) on 20 test sets of the datasets DBLP, WEBKB, CORA, and CITESEER. Italic indicates statistically significant best performance among the kernel methods and bold indicates statistically significant best performance among all methods both under a paired t-test ($p < 0.05$).

	5%							10%						
	CWK	DIFF	L+	VND	RL	LGC	LP	CWK	DIFF	L+	VND	RL	LGC	LP
DBLP	*62.8*	55.6	60.2	56.7	61.7	**64.0**	61.0	*69.3*	65.5	67.9	66.2	67.9	69.4	69.1
WEBKB	*61.5*	47.7	38.8	54.4	57.2	61.6	43.4	*63.2*	52.6	49.0	59.0	61.0	**65.6**	45.7
CORA	*72.1*	70.6	57.2	59.9	67.9	**73.8**	73.2	76.0	*77.2*	67.4	70.9	73.5	78.2	78.2
CITESEER	***53.5***	50.8	50.9	49.0	51.2	51.6	50.9	***57.8***	55.4	52.8	55.2	55.5	55.4	54.9



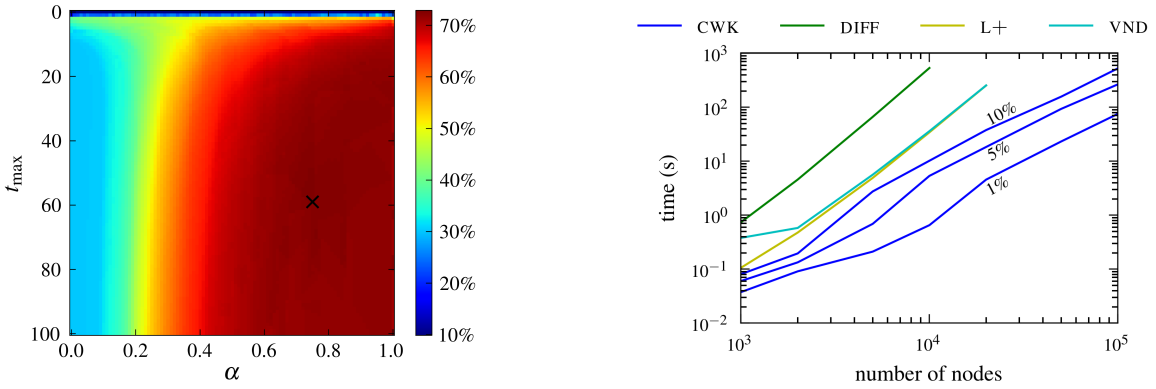
(a) WEBKB (b) CITESEER

Figure 3: **Average accuracies (and standard errors) on WEBKB and CITESEER.** The dotted lines indicate 5% and 10% training data, corresponding to the results reported in Table 3. Best viewed in colour.

4.3. Parameter Analysis and Runtimes

To analyse the sensitivity of CWK's predictive power with respect to changes in the kernel parameters, we computed the average accuracies over 10 randomly generated test sets for all combinations of α and $t_{\max}$, where $\alpha \in \{0.0, 0.01, \ldots, 1.0\}$ and $t_{\max} \in \{0, 1, \ldots, 100\}$ on the CORA dataset with 5% labeled data. A heatmap of the results is shown in Figure 4(a). Whereas the highest accuracy (72.1%) is achieved for an absorbing probability of $\alpha = 0.75$ and a maximum walk length of $t_{\max} = 59$, we see that for all $\alpha > 0.4$ and $t_{\max} > 2$, the accuracy is higher than 65% .This shows that CWK is not eminently sensitive to its parameters. The slight slope to the isoperformance curves suggest that walks of a given length and absorbing probability behave somewhat like slightly longer walks with a slightly smaller absorbing probability, which agrees with intuition.

In the following we analyse the scalability of the CWK computation for sparsely labeled networks. As investigating fast and scalable kernel methods goes beyond the scope of our work, we focus our analysis on the scalability of the kernel computation. We compare the runtimes for calculating all tested kernels on the POPULATED-PLACES dataset



(*a*) Heatmap (*b*) Runtimes

Figure 4: Panel (a) shows a heatmap of average accuracies w.r.t. $\boldsymbol{t}_{\max}$ and $\boldsymbol{\alpha}$. The accuracies are averaged over 10 randomly generated test sets of the CORA dataset with 5% labeled nodes. $\times$ marks the highest accuracy for $t_{\max} = 59$ and $\alpha = 0.75$. Panel (b) shows the loglog plot of runtimes for the kernel computations of all kernels on graphs (CWK with 1%, 5% and 10% labeled nodes, DIFF, L+, and VND) on PP-xk where $x \in \{10^3, 2 \times 10^3, 5 \times 10^3, \ldots, 10^5\}$. Best viewed in colour.

with up to 100 000 nodes. Once the kernel matrix is computed, all kernel-based methods scale comparably. Note that the iterative LP method not having to compute a kernel is usually faster then kernel-based classification; however, prediction results are also significantly worse for most datasets. Figure 4(b) shows the runtimes on the PP-xk dataset for $x \in \{1, 2, 5, 10, 20, 50, 100\}$. For CWK, whose runtime depends on the training fraction, we show curves for 1%, 5%, and 10% labeled nodes. We note that the CWK took about the same amount of time with $n = 100\,000$ as DIFF did for $n = 10\,000$ and L+ and VND did for $n = 20\,000$. The smaller slope for the CWK also shows a more slowly growing runtime in general. Finally, we make two remarks regarding the RL method, whose runtime is on a similar order as CWK's. First, RL must recalculate the kernel matrix multiple times, whereas we only compute it once. Second, the time and storage requirements of RL grow with the test size rather than the training size. For example, on the PP-100k dataset with 5% training data, CWK requires approximately 3.7 GB of storage, whereas RL requires about 71 GB.

5. Conclusions

In this paper, we introduced a new kernel on graphs, the coinciding walk kernel, bringing together graph-based label inference and kernel methods to leverage benefits from both fields for learning tasks in sparsely labeled networks. The kernel values of CWKs are given by the probability that the labels encountered during parallel absorbing random walks on partially labeled graphs coincide. That is, two nodes have a high kernel value if the labels surrounding each node are arranged similarly. Our extensive experiments demonstrated that CWK, which takes both Hypotheses 1 and 2 into account, scales and is robust across all tested datasets for node-label prediction in sparsely labeled graphs.

The probabilistic random walk design of CWKs leads to exciting ideas for future work. For example, we aim to use partially absorbing random walks for the design of a graph kernel for graph classification and retrieval. Further, we plan to investigate active learning with the CWK to improve even more upon the prediction results in sparsely-labeled networks.

Finally, hybrid semi-supervised support vector machines also constitute a great framework to investigate the power of CWKs in semi-supervised learning.

Acknowledgments

This work was supported by the European Commission under "FP7-248258-First-MM", the German Federal Office for Agriculture and Food (BLE) under "2815411310", the German Science Foundation (DFG) under "GA 1615/1-1", and the Fraunhofer ATTRACT grant STREAM.

References

O. Chapelle, V. Sindhwani, and S. S. Keerthi. Optimization Techniques for Semi-Supervised Support Vector Machines. *Journal of Machine Learning Research*, 9:203–233, 2008.

C. Desrosiers and G. Karypis. Within-Network Classification Using Local Structure Similarity. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD-09)*, pages 260–275, 2009.

F. Fouss, K. Françoisse, L. Yen, A. Pirotte, and M. Saerens. An Experimental Investigation of Kernels on Graphs for Collaborative Recommendation and Semisupervised Classification. *Neural Networks*, 31:53–72, 2012.

B. Gallagher, H. Tong, T. Eliassi-Rad, and C. Faloutsos. Using Ghost Edges for Classification in Sparsely Labeled Networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-08)*, pages 256–264, 2008.

T. Gärtner, P. Flach, and S. Wrobel. On Graph Kernels: Hardness Results and Efficient Alternatives. In *Computational Learning Theory and Kernel Machines — Proceedings of the 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop (COLT/Kernel-03)*, pages 129–143, 2003.

T. Hwang and R. Kuang. A Heterogeneous Label Propagation Algorithm for Disease Gene Discovery. In *Proceedings of the SIAM International Conference on Data Mining (SDM-10)*, pages 583–594, 2010.

M. Ji and J. Han. A Variance Minimization Criterion to Active Learning on Graphs. *Journal of Machine Learning Research - Proceedings Track (AISTATS-12)*, 22:556–564, 2012.

H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized Kernels Between Labeled Graphs. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML-03)*, pages 321–328, 2003.

R. I. Kondor and J. D. Lafferty. Diffusion Kernels on Graphs and Other Discrete Input Spaces. In *Machine Learning, Proceedings of the Nineteenth International Conference (ICML-02),*, pages 315–322, 2002.

F. Lin and W. W. Cohen. Semi-Supervised Classification of Network Data Using Very Few Labels. In *International Conference on Advances in Social Networks Analysis and Mining (ASONAM-10)*, pages 192–199, 2010a.

F. Lin and W. W. Cohen. Power Iteration Clustering. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 655–662, 2010b.

M. R. Min, A. J. Bonner, and Z. Zhang. Modifying Kernels Using Label Information Improves SVM Classification Performance. In *The Sixth International Conference on Machine Learning and Applications (ICMLA-07)*, pages 13–18, 2007.

M. Neumann, N. Patricia, R. Garnett, and K. Kersting. Efficient Graph Kernels by Randomization. In *Machine Learning and Knowledge Discovery in Databases - European Conference (ECML/PKDD-12)*, pages 378–393, 2012.

M. Neumann, R. Garnett, and K. Kersting. Coinciding Walk Kernels. In *Eleventh Workshop on Mining and Learning with Graphs (MLG-13)*, 2013.

J. Neville and D. Jensen. Leveraging Relational Autocorrelation with Latent Group Models. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM-05)*, pages 322–329, 2005.

P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective Classification in Network Data. *AI Magazine, Vol. 29, Nr. 3*, 29(3):93–106, 2008.

X. Shi, Y. Li, and P. S. Yu. Collective Prediction with Latent Graphs. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM-11)*, pages 1127–1136, 2011.

V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the Point Cloud: from Transductive to Semi-supervised Learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML-05)*, pages 824–831, 2005.

A. Smola and R. I. Kondor. Kernels and Regularization on Graphs. In *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop (COLT/Kernel-03)*, pages 144–158, 2003.

M. Szummer and T. Jaakkola. Partially Labeled Classification with Markov Random Walks. In *Advances in Neural Information Processing Systems (NIPS-01)*, pages 945–952, 2001.

X.-M. Wu, Z. Li, A. M.-C. So, J. Wright, and S.-F. Chang. Learning with Partially Absorbing Random Walks. In *Advances in Neural Information Processing Systems (NIPS-12)*, pages 3086–3094, 2012.

D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with Local and Global Consistency. In *Advances in Neural Information Processing Systems (NIPS-03)*, pages 321–328, 2003.

X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML-03)*, pages 912–919, 2003.

X. Zhu, J. S. Kandola, Z. Ghahramani, and J. D. Lafferty. Nonparametric Transforms of Graph Kernels for Semi-Supervised Learning. In *Advances in Neural Information Processing Systems (NIPS-04)*, pages 1641–1648, 2004.