Coinciding Walk Kernels

Marion Neumann
BIT, University of Bonn
Bonn, Germany
marion.neumann@uni-bonn.de

Roman Garnett
CS, University of Bonn
Bonn, Germany
rgarnett@uni-bonn.de

Kristian Kersting
IGG, University of Bonn
Bonn, Germany
kersting@igg.uni-bonn.de

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 state-of-the-art kernel- and walk-based methods on the task of node-label prediction in sparsely labeled graphs. 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 network of interlinked populated places extracted from the DBpedia knowledge base.

1. INTRODUCTION

The study of structure in networked data has led to great developments and success in graph-based and collective learning [14]. 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 [13]. 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 in the graph are likely to have the same label.

Eleventh Workshop on Mining and Learning with Graphs. Chicago, Illinois, USA

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 relations to accurately predict class labels. Previous work in this direction has introduced latent graphs by adding additional edges [15, 4], run multiple random walks with restarts [11], and suggested schemes for active learning [6].

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.

In this work, we encode the "label structure" surrounding a node by the sequences of labels encountered during random walks from that node. The main contribution is a new kernel, the *coinciding walk kernel* (CWK), that uses these walks to quantify how similarly the labels surrounding each node are arranged. A similar approach to one presented here, also incorporating local structure similarity, is introduced in [2]. In this relaxation labeling approach a similarity measure based on parallel RWs with constant termination probability is used. Besides having more free parameters and a higher space complexity, CWKs outperform this method on various datasets as we will show in our experimental evaluation.

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).[1] 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 [5, 7, 12]; the last reference used this idea to design a kernel among partially labeled graphs.

[1]We make this distinction between "kernels on graphs" and "graph kernels" throughout.

CWKs are inspired by the construction of these graph kernels; however, they define a kernel among the nodes of a graph and hence can be used for learning tasks like node-label prediction in partially labeled networks. Common kernels on graphs include the diffusion kernel [8], the p-step random walk kernel [17], and the Moore–Penrose pseudoinverse of the Laplacian, L^+, [3] which is a limiting case of the regularized Laplacian kernel. 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 [21, 19]. The CWK uses this idea to define a kernel between nodes of a partially labeled graph. 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 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 [21, 19, 3, 8, 11]. However, RWs using the graph's adjacency matrix as the transition matrix converge to a constant steady-state distribution. Thus, the idea of early stopping was successfully introduced in power iteration methods for clustering [10] and node-label prediction [18]. 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, making the CWK a data-dependent kernel. Data-dependent kernels are widely used in semi-supervised learning [16, 9], 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 [1] 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 for plain graph data where no feature information on the nodes is given, in contrast to the standard semi-supervised learning setting. However, both approaches could complement each other under the right circumstances.

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 contributions of this paper are:

- introducing the coinciding walk kernel, that is,
- developing the first kernel on graphs leveraging label information in kernel construction, and
- providing a 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, we discuss its probabilistic interpretation and show its positive definiteness. Before concluding, we present experimental results on several state-of-the-art graph datasets.

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 model label-structure similarity by partially label-absorbing random walks.

Before developing the technical details, let us first provide an intuition. Consider a particle traveling from node to node via the edges of a graph such that the decision of where to go next only depends on its current location. This is a Markov random walk on a graph. We can modify this behaviour by introducing *absorbing states* – when reaching an absorbing state, our particle is not able to continue its walk, but is instead caught in a loop staying at that node. The absorbing states could for example be chosen as the subset of the nodes having observed labels, a natural choice in our paradigm. Further, we can define partially absorbing random walks, where absorbing nodes only activate with a given probability. Now, we will give the technical definitions needed for defining and understanding coinciding walk kernels.

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 $A \in \mathbb{R}^{n \times n}$. A random walk on G is a Markov process $X = \{X_t : t \geq 0\}$ on $V = \{1, 2, ..., n\}$ 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

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

The nodes in S are "absorbing" in that the 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.[2] 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 [20] can be cast in terms of simulating absorbing 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) [19].

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 [18, 10]. 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]This prior could also be the output of an external classifier built on available node attributes.

2.3 Parallel Absorbing Random Walks

The final ingredient we need are parallel[3] 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.

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.

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 $\tilde{P}_t$, i.e.,

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

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

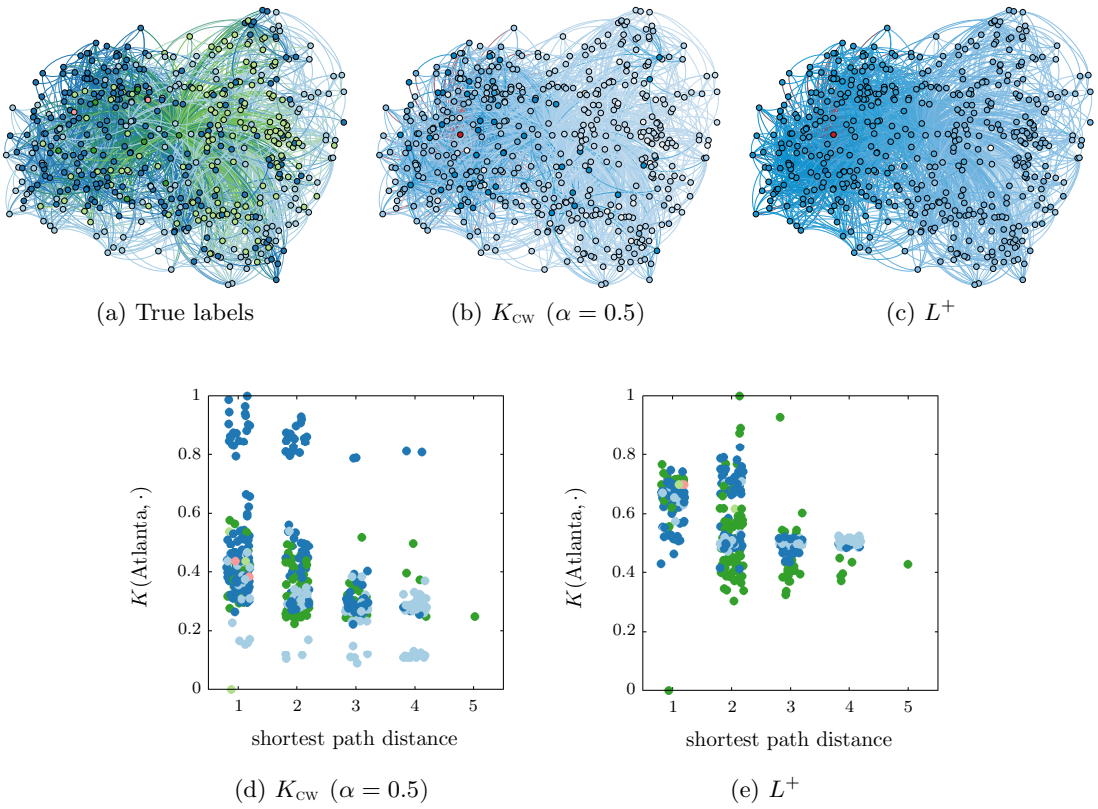
Note that $\tilde{P}_t \in \mathbb{R}^{n+k\times k}$ is simply the probability matrix P_t extended by a $k \times k$ identity matrix. By using finite-length PARWs K_{CW} has two kernel parameters: the absorbing probability α, and the maximum walk length $t_{\max}$, where α controls trade-off between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

$(P_t)_i (P_t)_j^\top$ 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 random walk 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).

[3]Note that we do not use the term "parallel" in the context of parallel computing.



(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) illustrates the values of the coinciding walk kernel (row of K_{CW}) for Atlanta, coloured red. Dark blue means high similarity, i.e., high kernel value, and white represents low similarity. Panel (c) illustrates the values of L^+ for Atlanta. Panels (d) and (e) show scatter plots of the shortest path distance vs. the normalized values of $K(\textbf{Atlanta}, \cdot)$ for K_{CW} and L^+ respectively, where the colours encode the class labels. Best viewed in colour.

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., $K_{\text{CW}}(i, j) \propto \sum_{t=0}^{t_{\max}} (P_t)_i (P_t)_j^\top$.

The computation of CWK on a graph G with adjacency matrix A given the initial label distributions P_0 and its parameters is summarized in Algorithm 1. The computational complexity of the naive calculation is $\mathcal{O}(k\, t_{\max}\, n^2)$. However, 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. Hence, the complexity for the kernel computation is $\mathcal{O}(k\, t_{\max}\, |V_L| n)$ which is significantly lower for sparsely labeled graphs as $|V_L| \ll n$.

In Figure 1 we provide an illustration of CWK on a subgraph of a labeled graph built from concepts in the DBpedia[4] ontology marked as "populated places." 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 (obeying the homophily assumption); whereas the value of K_{CW} also shows some highly correlated far-away

[4] www.dbpedia.org

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}

$K \leftarrow P_0 P_0^\top$
$T = D^{-1}A$, where $D = \text{diag}(\sum_j A_{ij})$
$\tilde{T} \leftarrow \text{constructTrans}(T, \alpha)$ (cf. Eq. (3))
for $t \leftarrow 1...t_{\max}$ **do**
  $\tilde{P}_t \leftarrow \tilde{T}\tilde{P}_{t-1}$ (one step transition)
  $P_t \leftarrow (\tilde{P}_t)_i,\ i \in \{1,..,n\}$
  $K \leftarrow K + P_t P_t^\top$ (add kernel contribution)
end for
$K_{\text{CW}} \leftarrow \frac{1}{t_{\max}+1}\ K$ (normalize kernel)

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.

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. We compare their performance to existing methods from the kernel and collective inference community. The two main questions to answer here are whether CWKs are able to utilize both autocorrelation and structure similarity for node-label classification and whether employing CWKs improves over state-of-the-art graph-based learning methods.

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 [20]
- RL: relaxation labeling using structure similarity [2]
- DIFF: diffusion kernel [8]
- L+: Moore–Penrose pseudoinverse of the Laplacian kernel [3].

LP is the obvious baseline approach. Further, we choose DIFF and L+ to represent existing successful kernels on graphs, and RL as it is currently the most accurate method in the area of collective classification, c.f. results in [2]. RL is the closest approach to CWKs, also incorporating local structure similarity. The used similarity measure is based on parallel RWs with constant termination probability in a relaxation labeling algorithm.

The following graph datasets are used for evaluation:

- DBLP[5] (connected coauthor graph extracted from the DBLP database)
- WEBKB[6] (cocitation graph of webpages from computer science departments of four universities)

[5] www.cs.illinois.edu/homes/mingji1/DBLP_four_area.zip
[6] www.netkit-srl.sourceforge.net/data.html

Table 1: Dataset properties. PP-xK is short for POPULATED-PLACES-xK, where x is 1, 3, or 5. The last column indicates the percentage of the most frequent class.

	properties		
dataset	# nodes	# labels	% of most frequent class
DBLP	1 711	4	36%
WEBKB	1 462	6	28%
CORA	2 708	7	30%
CITESEER	3 264	6	21%
PP-xK	x	5	49% (avg)

- CORA[7] (citation network of scientific papers)
- CITESEER[7] (citation network of scientific papers)
- POPULATED-PLACES (link graph extracted from DBpedia, described above).

The properties of the datasets are summarized in Table 1. For WEBKB we used the cocitation networks of all four universities (Cornell, Texas, Washington, and Wisconsin), combined into one (disconnected) graph. For the POPULATED-PLACES dataset, we created graphs of varying sizes by performing a breadth-first search from the first node in the graph (Alabama).

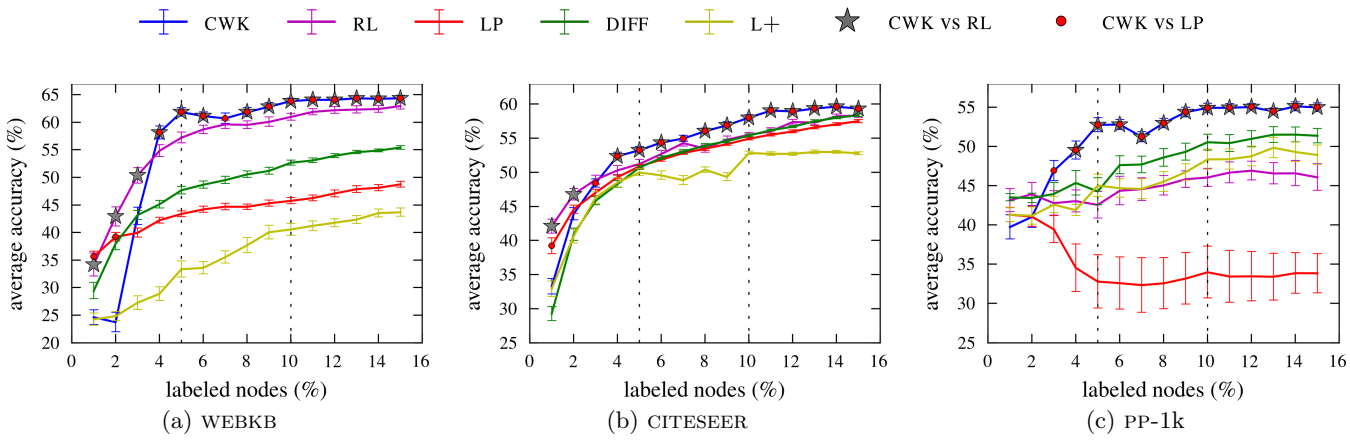
We focus on sparsely labeled graphs and use 20 randomly generated test splits for 1% up to 15% labeled nodes. The test sets are the same for each method and all reported classification accuracies are an average over the results on the respective 20 test sets. The classification performance of all kernel-based methods is evaluated by running C-SVM classifications using `libSVM`.[8] Parameter learning of all compared methods is done by the following protocol. For each method we train all parameters (including the SVM cost parameter of the kernel-based methods) jointly via grid search on 10 randomly generated training splits having 5% and 10% labeled nodes, respectively. 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: $t_{\max} \in \{0, 1, \ldots, 10, 20, \ldots, 200\}$, $\alpha \in \{0, 0.05, 0.1, 0.2, 0.5, 0.8, 0.9, 0.95, 1\}$
- RL: $N \in \{1, 2, \ldots, 5\}$, $\gamma \in \{0.1, 0.3, 0.5, 0.7\}$, $\alpha_{\text{RL}} \in \{0.25, 0.5, \ldots, 1.5\}$, $\beta_{\text{RL}} \in \{0.5, 1.0, \ldots, 3.0\}$
- DIFF: $\beta \in 2^{\{-4,-3,\ldots,7\}}$
- all kernel methods: SVM cost $C \in 2^{\{-4,-3,\ldots,7\}}$.

4.2 Predictive Performance

The predictive performances for all datasets with 5% and 10% labeled nodes are summarized in Table 2. In the scenario with 5% labeled nodes, CWK performed significantly better (under a paired t-test with $p < 0.05$) than the comparing methods on four out of seven datasets. On two of the remaining datasets (CORA and PP-3k), CWK achieved the second-best average accuracy. When considering 10% la-

[7] www.cs.umd.edu/projects/linqs/projects/lbc/index.html
[8] www.csie.ntu.edu.tw/~cjlin/libsvm/



(a) WEBKB (b) CITESEER (c) PP-1k

Figure 2: Average accuracies (and standard errors) for WEBKB, CITESEER, and PP-1k. Accuracies are averaged over 20 randomly generated test splits of 1% to 15% labeled nodes for coinciding walk kernel (CWK), label propagation (LP), relaxation labeling using structure similarity (RL), diffusion kernel (DIFF), and pseudo inverse of the Laplacian (L+). The error bars indicate standard error over the 20 test sets. The star refers to a statistically significant difference between CWK and RL; the red dot indicates a statistically significant difference of CWK and LP ($p < 0.05$). The dotted lines indicate 5% and 10% training data corresponding to the results reported in Table 2. Best viewed in colour.

Table 2: Average accuracies (%) for all tested methods on all datasets using 5% and 10% labeled nodes. Accuracies are averaged over 20 randomly generated test sets. Bold indicates best method and • statistically significant difference to the respective second best method under a paired t-test ($p < 0.05$). PP-xk is short for POPULATED-PLACES-xk.

	5%					10%				
	CWK	RL	DIFF	L^+	LP	CWK	RL	DIFF	L^+	LP
DBLP	**62.9**•	61.7	55.5	59.9	61.2	**69.2**	67.9	65.4	67.7	69.1
WEBKB	**61.9**•	57.2	47.7	33.4	43.4	**63.9**•	61.0	52.7	40.6	45.8
CORA	72.1	67.9	70.4	57.1	**73.2**•	76.1	73.5	77.1	67.2	**78.2**•
CITESEER	**53.2**•	51.2	50.7	50.0	50.9	**58.0**•	55.5	55.4	52.8	54.9
PP-1k	**52.8**•	42.6	44.3	45.0	32.8	**54.9**•	46.1	50.6	48.4	34.0
PP-3k	59.5	57.7	**60.2**	51.4	45.6	**63.7**	59.4	63.1	53.7	50.3
PP-5k	**60.9**	53.3	59.6	53.3	40.2	**63.5**•	59.2	61.7	55.2	40.5

beled nodes, CWK performed better in five out of seven cases, four of which were statistically significant. Overall, CWK is performing significantly better than all baseline approaches according to a McNemar's test ($p < 0.05$). Figure 2 shows plots of the average accuracies of all compared methods for 1% up to 15% labeled nodes for the WEBKB, CITESEER, and PP-1k datasets. CWK generally outperforms the other methods. For the WEBKB dataset in particular, leveraging label-structure similarity yields a huge benefit, as both CWK and RL perform considerably better than the other methods. It is not surprising that LP fails to accurately predict the labels in the populated places graphs, as it relies purely on the homophily assumption (Hypothesis 1) and fails to leverage the structure similarity inherent to these networks. Surprisingly, RL does not achieve better results than the compared kernels on graphs DIFF and L+ on these datasets either.

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 using CWKs, i.e., taking both Hypotheses 1 and 2 into account, considerably improves node-label prediction in sparsely labeled graphs.

6. ACKNOWLEDGMENTS

This work was supported by the European Commission under contract "FP7-248258-First-MM", by the Fraunhofer

ATTRACT fellowship STREAM, and by the German Science Foundation (DFG) under reference number "GA 1615/1-1".

7. REFERENCES

[1] 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.

[2] 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.

[3] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):355–369, 2007.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] G. Lever, T. Diethe, and J. Shawe-Taylor. Data Dependent Kernels in Nearly-linear Time. *Journal of Machine Learning Research - Proceedings Track (AISTATS-12)*, 22:685–693, 2012.

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

[11] 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, 2010.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] M. Szummer and T. Jaakkola. Partially Labeled Classification with Markov Random Walks. In *Proceedings of Annual Conference on Neural Information Processing Systems (NIPS-01)*, pages 945–952, 2001.

[19] X.-M. Wu, Z. Li, A. M.-C. So, J. Wright, and S.-F. Chang. Learning with Partially Absorbing Random Walks. In *Proceedings of Annual Conference on Neural Information Processing Systems (NIPS-12)*, pages 3086–3094, 2012.

[20] X. Zhu and Z. Ghahramani. Learning from Labeled and Unlabeled Data with Label Propagation. Technical report, CMU-CALD-02-107, Carnegie Mellon University, 2002.

[21] 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.