Directed Probabilistic Watershed

Anonymous Author(s)
Affiliation
Address
email

Abstract

The Probabilistic Watershed is a semi-supervised learning algorithm applied on undirected graphs. Given a set of labeled nodes (seeds), it defines a Gibbs probability distribution over all possible spanning forests disconnecting the seeds. It calculates, for every node, the probability of sampling a forest connecting a certain seed with the considered node. We propose the "Directed Probabilistic Watershed", an extension of the Probabilistic Watershed algorithm to directed graphs. Building on the Probabilistic Watershed, we apply the Matrix Tree Theorem for directed graphs and define a Gibbs probability distribution over all incoming directed forests rooted at the seeds. Similar to the undirected case, this turns out to be equivalent to the Directed Random Walker. Furthermore, we show that in the limit case in which the Gibbs distribution has infinitely high temperature, the labeling of the Directed Probabilistic Watershed is equal to the one induced by the incoming directed forest of minimum weight. Finally, for illustration, we show the empirical performance of the proposed method in comparison with another semi-supervised segmentation method for directed graphs.

1 Introduction

In the last years, machine learning has experienced a great boost thanks to the large quantity of available data and increasing availability of less expensive and more powerful processing capacities. However, the acquisition of labeled data is still laborious in some applications like computer-aided diagnosis or drug discovery. Semi-supervised learning is the subfield of machine learning that utilizes both labeled and unlabeled data. It permits exploiting the large amount of unlabeled data available in many use cases jointly with frequently smaller sets of labeled data. When data is encoded as a network, graph-based semi-supervised learning aims to assign a label or class to each of the nodes of a network based on its topology and a given set of labeled nodes/seeds. Graph-based semi-supervised learning has been applied in multiple domains like computer vision [6, 15, 16], NLP [2, 27, 29], social networks [1, 41] and biomedical science[23]. Most graph-based semi-supervised learning algorithms focus on undirected weighted graphs [5, 32], though directed graphs appear naturally in many cases like in k-Nearest Neighbors graphs or citation and recommendation networks among others. Instances of methods proposed for directed graphs are [9, 11, 38, 44, 45].

Recently, Fita Sanmartin et al. [12] proposed the transductive graph-based semi-supervised learning algorithm Probabilistic Watershed (ProbWS). It envisages a Gibbs distribution over the labeled nodes separating forests (forests whose trees contain a unique labeled node) that span a weighted undirected graph. The probability of each forest is proportional to its weight. By means of the Matrix Tree Theorem [18, 36], the Probabilistic Watershed computes the probability of sampling a seed separating forest such that a query node is connected to a single labeled / seed node. This approach turns out to be equivalent computationally and by result to the Random Walker / Harmonic energy minimization [3, 15, 43, 46].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

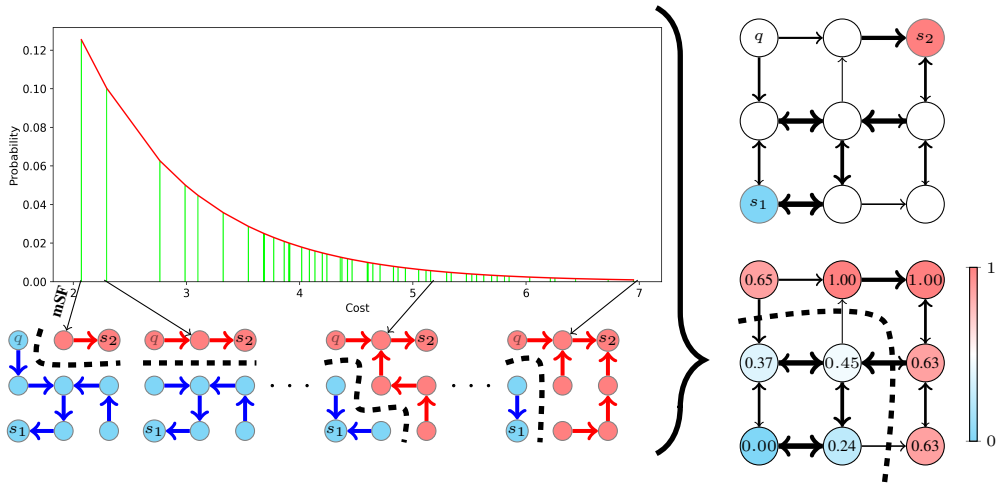


Figure 1: The DProbWS computes the expected seed assignment of every node for a Gibbs distribution over all exponentially many spanning in-forests in closed-form. It thus avoids the winner-takes-all behaviour of the minimum spanning forest (mSF). (**Top right**) Two-seeded directed graph with weighted edges represented by the widths of the arrows. (**Bottom left**) The mSF and other, higher cost in-forests. The mSF assigns the query node q to s_1. Other in-forests of low cost might however induce different labelings. (**Top left**) We therefore consider a Gibbs distribution over all spanning in-forests with respect to their cost (see equation (6)). Each green bar corresponds to the cost of one of the 60 possible spanning in-forests. (**Bottom right**) DProbWS probabilities for assigning a node to s_2. The dashed lines indicate the cut of the assignment. Query q is now assigned to s_2. Considering a distribution over all spanning forests gives an uncertainty measure and can yield a different assignment from the mSF's. In contrast to the 60 forests in this toy graph, for real-life graphs the number of in-forests increases exponentially with the number of edges and nodes.

The Matrix Tree Theorem [18, 36] has been generalized to directed graphs [24]. We propose an extension of the ProbWS, the "Directed Probabilistic Watershed" (DProbWS), that can be applied to directed graphs. Instead of considering seed separating spanning forests, we deal with incoming directed forests rooted at the seed nodes (see Figure 1). We analyze the Directed Probabilistic Watershed from a theoretical point of view and generalize the results presented in [12] to directed graphs. Building on the arguments used in the original Probabilistic Watershed paper, we

- analytically compute the probability that a graph node is assigned to a particular seed in an ensemble of Gibbs distributed incoming directed spanning forests rooted at the seeds,
- demonstrate the equivalence of the Directed Probabilistic Watershed and the directed version of the Random Walker presented in [15],
- propose a natural extension of the Power Watershed [6] to directed graphs,
- we run an exemplifying experiment to show the empirical performance of the DProbWS and compare it with the method proposed in [9].

1.1 Related work

Semi-supervised learning is an important subfield of machine learning due to is ability to exploit both labeled and unlabeled data [37]. If data is represented as a graph, graph based semi-supervised learning utilizes, in addition to the labeled samples, the topology of the graph to infer the class of the unlabeled nodes. Undirected graphs have received more attention on this field than the directed ones [5, 32]. Nonetheless, there have been methods proposed for directed graphs. In [44, 45] the labels are inferred by minimizing a regularized function that forces highly connected subgraphs to have the same label. Wang et al. [38] build a symmetric pairwise similarity via co-linkage analysis from the directed graph in order to apply algorithms for undirected graphs. The work presented in [11] proposes a method based on game theory, and the game's Nash equilibrium corresponds to the labeling of the data. Related to our work by the use of absorbing random walks is the one proposed in

[9]. Their algorithm assigns the label of the seed that maximizes the accumulated expected number of visits from the unlabeled query node before being absorbed by an artificially added metanode. We, instead, prove that the DProbWS turns out to assign the label of the seed that maximizes the probability of absorption of a random walker.

Our method extends the Probabilistic Watershed (ProbWS) [12], which, in turn, is based on the Watershed algorithm [7]. The ProbWS paper established a link between the Watershed [7] and the Random Walker [3, 15, 43, 46]. Another such connection was made by the Power Watershed paper [6]. We extend these relations to the directed graph framework.

We smooth the combinatorial minimum spanning in-forest via entropic regularization. More specifically, we consider a Gibbs distribution over all spanning in-forests, as was proposed in the original ProbWS paper [12]. Entropic regularization has been applied in other situations like the randomized shortest path framework [14, 19] or the optimal transport [8]. If applied in conjunction with deep networks, it allows end-to-end training [30].

The directed version of the Matrix Tree Theorem (MTT) [24] is key to our closed form computation of the weight of a set of incoming forests. The MTT has been also applied in NLP[21], biology [34] and network analysis [33, 35]. A generalization of the MTT, the Matrix Forest Theorem (MFT), was used by Chebotarev and Shamis [4] to define a distance between the nodes of a graph. Similar to our approach, Senelle et al. [31] define a Gibbs distribution over the forests using the MFT.

2 Background

2.1 Notation and terminology

In this section, we introduce the main definitions and notation used in the paper. Most of them have been borrowed from [12, 24]. Let $G = (V, E, w, c)$ be a directed graph where V denotes the set of nodes, E the set of edges and w and c are functions that assign a weight $w(e) \in \mathbb{R}_{\geq 0}$ and a cost $c(e) \in \mathbb{R}$ to each edge $e \in E$. The edge $e = (i, j)$ indicates that there is a directed edge from i to j. If $(i, j) \notin E$, we set $c\big((i, j)\big) = \infty$ and $w\big((i, j)\big) = 0$. All the graphs considered do not contain self-loops. In order to ease the exposition we only consider the 2-seed scenario, although the method can be generalized to the n-seed scenario. The set $S = \{s_1, s_2\} \subset V$ will denote the set of seed/labeled nodes and $U = V \backslash S$ the unlabeled ones. We will use the terms seed and labeled nodes interchangeably.

Definition 1. Let $G = (V, E, w)$ be a weighted directed graph. Given $v \in V$, we say that the subgraph $t = (V_t, E_t) \subseteq G$ is an **incoming directed spanning tree rooted at r** (spanning in-tree for short) if: **1)** $V_t = V$; **2)** every vertex $u \neq r$ has one and only one outgoing edge; **3)** the root node r does not have any outgoing edges; **4)** t does not have directed cycles, i.e. there does not exist a directed path in t such that the initial and final vertex are the same. Note that condition 4), in conjunction with the other conditions, implies that no cycles (directed and undirected) will be formed.

Equivalently, one can define an in-tree as a subgraph in which, for any vertex $u \in V$ there is exactly one directed path from u to the root r, and the root does not have any out-going edges.[1] Note that an in-tree becomes a tree if the directions of the edges are ignored. The set of in-trees rooted at r will be denoted by $\mathcal{T}^{\overrightarrow{r}}$.

The set $\mathcal{F}^{\overrightarrow{v}}_{\overrightarrow{u}}$ represents the set of 2-in-trees spanning forests rooted at u and v, i.e. the spanning forests consisting of two in-trees, such that u and v are the respective roots of the in-trees. Furthermore, if we consider a third node q, we define $\mathcal{F}^{\overrightarrow{v}}_{\overrightarrow{u},q} \subseteq \mathcal{F}^{\overrightarrow{v}}_{\overrightarrow{u}}$, as the set of all 2-in-trees spanning forests where q is connected to u by a directed path. In order to shorten the notation we will refer to 2-in-trees spanning forests simply as 2-in-forests or in-forests.

We define the weight of a graph as the product of the weights of all its edges, $w(G) = \prod_{e \in E} w(e)$. The weight of a set of graphs, $w(\{G_i\}_{i=0}^{n})$ is the sum of the weights of the graphs G_i. In a similar manner, we define the cost of a graph as the sum of the costs of all its edges, $c(G) = \sum_{e \in E} c(e)$. As in the Probabilistic Watershed (ProbWS) [12], we consider $w(e) = \exp(-\mu c(e))$, $\mu \geq 0$, which will

[1] One can define analogously an outgoing spanning tree (out-tree) rooted at r as a tree where there is exactly one directed path from the root r to any other node, and the root does not have any incoming edges.

be a consequence of the definition of a Gibbs distribution over the 2-in-forests in $\mathcal{F}_{\vec{u}}^{\vec{v}}$. Thus, a low edge-cost corresponds to a large edge-weight, and a minimum edge-cost spanning in-forest (mSF) is equivalent to a maximum edge-weight spanning in-forest (MSF). Depending on the context, the abbreviations mSF and MSF will be also used for the undirected versions of the minimum/maximum spanning forests. Via the Directed Matrix Tree Theorem, we aim to compute the weight of the set of the 2-in-forests rooted at the seeds. The theorem makes use of the out-Laplacian matrix which we define next.

Definition 2. Given a weighted directed graph $G = (V, E, w)$ we define the **out-Laplacian** of G as

$$L \coloneqq D - A^\top, \tag{1}$$

where $A \in \mathbb{R}^{|V|\times|V|}$ is the vertex-adjacency matrix of G represented entry-wise as $A_{ij} = w\big((i,j)\big)$ and D denotes the diagonal matrix defined as $D_{ii} = \sum_{j\in V} A_{ij}$, i.e. D_{ii} is the out-degree of vertex i.[2] For brevity, we will refer to the out-Laplacian just as Laplacian. For any $v \in V$, $L^{[v]}$ will stand for the Laplacian after removing the row and column indexed by v.

2.2 Matrix Tree Theorem

Theorem 2.1 (**MTTdir**, [24])**.** For any edge-weighted directed graph G with an arbitrary fixed node $r \in V$ the weight of incoming directed spanning trees rooted at r, $w(\mathcal{T}^{\vec{r}})$, is equal to[3]

$$w(\mathcal{T}^{\vec{r}}) \coloneqq \sum_{t\in\mathcal{T}^{\vec{r}}} w(t) = \sum_{t\in\mathcal{T}^{\vec{r}}} \prod_{e\in E_t} w(e) = \det(L^{[r]}).$$

Note that Theorem 2.1 generalizes the original Matrix Tree Theorem [18, 36] since any spanning tree in an undirected graph can be interpreted as an in-tree rooted at r after setting the direction of the edges appropriately. Hence, for an undirected graph the choice of the root r is irrelevant and we retrieve the original MTT.

We propose to follow the approach taken in [12], but with the generalization of the Matrix Tree Theorem to directed graphs and thus obtain a directed version of the ProbWS.

2.3 Random Walker and the Power Watershed

In this section, we summarize the Random Walker [15] and Power Watershed [6], two semi-supervised graph-based algorithms which are directly connected to the ProbWS. Both algorithms consider an undirected graph, G, with a set of seeds $S = \{s_1, s_2\}$, i.e. labeled nodes.

The Random Walker paper [15] addresses the following problem: *What is the probability that a random walker starting at node q reaches seed s_1 before reaching s_2?* The solution to this question can be obtained by solving the combinatorial Dirichlet problem, which consists in finding the minimizer of

$$\frac{1}{2}x^\top L x = \frac{1}{2}\sum_{e=\{u,v\}\in E} w(e)(x_u - x_v)^2, \text{ s.t. } \quad x_{s_1} = 1,\ x_{s_2} = 0, \tag{2}$$

where L denotes the Laplacian of G. The minimizer of equation (2) is the solution of the following linear system

$$L_U x_U^{s_i} = -B_{s_i}^\top, \tag{3}$$

where $x_U^{s_i}$ represents the probability that the unlabeled nodes in U are absorbed by s_i, L_U is the square submatrix of L indexed by the elements in U and $B_{s_i}^\top$ is the column s_i of L without the seeds.

The Power Watershed generalizes the framework of the Random Walker and minimizes the following objective function

$$\sum_{e=\{u,v\}\in E} w^\alpha(e)(x_u - x_v)^\beta, \text{ s.t. } \quad x_{s_1} = 1,\ x_{s_2} = 0, \tag{4}$$

[2] Analagously, the in-Laplacian is defined as $L_{in} = D_{in} - A$, where D_{in} is the diagonal matrix with the in-degrees of the nodes.

[3] The weight of the out-trees can be calculated using the in-Laplacian instead of the out-Laplacian.

for $\alpha, \beta \geq 0$. For instance, $\alpha = 1$ and $\beta = 2$ define the Random Walker's objective function. The Power Watershed analyzes the case $\alpha \to \infty$ and $\beta = 2$, and boils down to computing a mSF (minimum Spanning Forest) and applying the Random Walker in the plateaus (connected subgraphs with constant edge weight), resolving the ambiguity of which mSF is sampled when there is more than one.

The generalization of these algorithms to directed graphs are not an easy task if one takes the objective function approach. If the graph is directed, then the Laplacian matrix is not symmetric anymore, and therefore is not ensured that it will be positive semi-definite. Hence, we can not guarantee the existence of a minimizer for the combinatorial Dirichlet problem (2).

The Random Walker can actually be generalized to the directed case if we use the intuitive approach instead, i.e. we consider the probability of a random walker (constrained to follow the directions of the edges) being absorbed by a certain seed. In the supplemental material, we give a proof of how these probabilities can be computed in the directed case. We demonstrate that the solution of the linear system (3) with the Laplacian transposed provides the desired probabilities. In section 5, we show how the Power Watershed can be generalized to directed graphs by means of the DProbWS.

2.4 Probabilistic Watershed review

The Probabilistic Watershed (ProbWS) [12] is based on the Watershed algorithm. The Watershed algorithm calculates a minimum cost spanning forest, mSF, such that the seeds belong to different components [7]. A query node inherits the label of the seed to which it is connected in the mSF. The ProbWS instead, considers all possible seed-separating spanning forests. It defines a Gibbs probability distribution over the forests and analytically computes, for every node, the probability of sampling a forest connecting a certain seed with that node via the application of the Matrix Tree Theorem (for undirected graphs).

Fita Sanmartin et al. [12] proved that the ProbWS turns out to be equivalent to the Random Walker algorithm proposed by Grady [15]. Additionally, it was shown that the Power Watershed potentials [6] were given by the ProbWS probabilities when the latter is restricted to mSFs instead of all spanning forests. This restriction manifests when the entropy of the Gibbs distribution of forests is minimized.

In this work, we extend the ProbWS framework to directed graphs and call it Directed Probabilistic Wasterhsed. In this case, the spanning forests considered are formed by in-forests rooted at the seeds, i.e. directed trees with a unique path from every node to the seed. Inspired by [12], we define a Gibbs Probability distribution over the in-forests and we apply the Matrix Tree Theorem for directed graphs to compute the probability of sampling a forest connecting a certain seed with a query node in closed-form. Next, we demonstrate the equivalence to the Directed Random Walker and finally we propose an extension of the Power Watershed for directed graphs.

3 Directed Probabilistic Watershed

3.1 Gibbs probability distribution

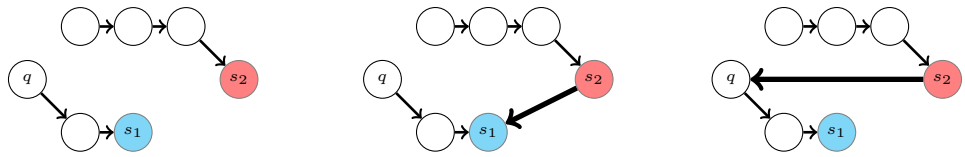
As in [12], we define a Gibbs probability distribution over 2-in-forests with given entropy J, such that the 2-in-forests with a lower cost have a higher probability mass. Formally, the 2-forests are sampled from the distribution which minimizes

$$\min_P \sum_{f \in \mathcal{F}_{\vec{s_1}}^{\vec{s_2}}} P(f)c(f), \quad \text{s.t.} \quad \sum_{f \in \mathcal{F}_{\vec{s_1}}^{\vec{s_2}}} P(f) = 1 \text{ and } \mathcal{H}(P) = J, \tag{5}$$

where $\mathcal{H}(P)$ is the entropy of P. The lower the entropy, the more probability mass is given to the 2-in-forests of lowest cost. The minimizing distribution is the Gibbs distribution ([see 39, 3.2]):

$$P(f) = \frac{\exp\left(-\mu c(f)\right)}{\sum_{f' \in \mathcal{F}_{\vec{s_1}}^{\vec{s_2}}} \exp\left(-\mu c(f')\right)} = \frac{\prod_{e \in E_f} \exp(-\mu c(e))}{\sum_{f' \in \mathcal{F}_{\vec{s_1}}^{\vec{s_2}}} \prod_{e' \in E_{f'}} \exp(-\mu c(e'))} = \frac{w(f)}{\sum_{f' \in \mathcal{F}_{\vec{s_1}}^{\vec{s_2}}} w(f')}, \tag{6}$$

where μ implicitly determines the entropy. A higher μ implies a lower entropy. Equation (6) motivates our choice of edge weights $w(e) = \exp(-\mu c(e))$.



(a) $f \in \mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1},q} \subset \mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1}}$ (b) $t \in \mathcal{T}^{\overrightarrow{s_1}}_{\bar{G}_1}$ (c) $t \in \mathcal{T}^{\overrightarrow{s_1}}_{\bar{G}_2}$

Figure 2: Procedure to transform a forest in $\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1},q}$ to a spanning tree in $\bar{G}_1$ and $\bar{G}_2$ as defined in Lemma 3.3. (2a) Forest in $f \in \mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1},q} \subseteq \mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1}}$. (2b) Transformation of f into a tree in $t \in \mathcal{T}^{\overrightarrow{s_1}}_{\bar{G}_1}$ after adding the edge (s_2, s_1) (thick edge). (2c) Transformation of f into a tree in $t \in \mathcal{T}^{\overrightarrow{s_1}}_{\bar{G}_2}$ after adding the edge (s_2, q) (thick edge).

Definition 3 (**Probabilities of the Directed Probabilistic Watershed**). Given two seeds s_1 and s_2 and a query node q, we define the Directed Probabilistic Watershed's probability that q and s_1 have the same label as the probability of sampling a 2-in-forest from $\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1}}$ that connects q to s_1 by a directed path, i.e. they belong to the same in-tree rooted at s_1:

$$P(q \sim s_1) := \sum_{f \in \mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1},q}} P(f) = \sum_{f \in \mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1},q}} w(f) \Big/ \sum_{f' \in \mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1}}} w(f') = w\left(\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1},q}\right) / w\left(\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1}}\right). \quad (7)$$

The Directed Probabilistic Watershed (DProbWS) takes all spanning 2-in-forests into account according to their cost (see Figure1). The resulting assignment probability of each node provides an uncertainty measure. Assigning each node to the seed for which it has the highest probability can yield a labeling different from the forest with highest individual probability, the mSF.

Remark 3.1. Note that the probability given by the DProbWS is well defined if and only if $w\left(\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1}}\right) \neq 0$, that is if and only if there exists at least one in-forest rooted at the seeds. As far as any node is connected by a directed path to at least one of the seeds, such a forest will exist. If the seeds were not reachable from a node q, this node would have no connection with the seeds and therefore we could not infer any label. These nodes are termed zero-knowledge nodes. We can still use the proposed method by removing all nodes in the zero-knowledge components [28]. In the remaining we will assume $w\left(\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1}}\right) \neq 0$.

Remark 3.2. Note that the DProbWS probabilities have been based on the concept of incoming forests. The same reasoning can be extended for outgoing forests just by reversing the edge directions.

3.2 Computation of the Directed Probabilistic Watershed probabilities

In this section, we explain how to compute the probabilities defined by the DProbWS. While for undirected graphs it is possible to compute $w(\mathcal{F}^{s_2}_{s_1})$ as a difference of the weight of the set of spanning trees of the graph G and the weight of the set of spanning trees of G after adding an edge connecting the seeds, this is not possible for $w(\mathcal{F}^{s_2}_{s_1,q})$.[4] In [12], $w(\mathcal{F}^{s_2}_{s_1,q})$ is obtained by solving a linear system involving $w(\mathcal{F}^{s_2}_{s_1})$, $w(\mathcal{F}^{q}_{s_2})$ and $w(\mathcal{F}^{q}_{s_1})$. In contrast to the undirected case, we can bypass the auxiliary linear system due to the directions of the edges which restrict the number of possible directed forests. This permits us to express $w(\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1}})$ and $w(\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1},q})$ in terms of the weight of the set of in-trees of an augmented graph as it is shown in the following lemma.

Lemma 3.3. Let $G = (V, E, w)$ be a directed graph. Given $s_1, s_2, q \in V$, define the graphs $\bar{G}_1 = (V, E \cup (s_2, s_1), \bar{w}_1)$ and $\bar{G}_2 = (V, E \cup (s_2, q), \bar{w}_2)$ where $\bar{w}_1(e) = \bar{w}_2(e) = w(e)\ \forall e \in E$ and $\bar{w}_1\big((s_2, s_1)\big) = \bar{w}_2\big((s_2, q)\big) = 1$. Then

$$\textbf{a)}\ w\left(\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1}}\right) = w\left(\mathcal{T}^{\overrightarrow{s_1}}_{\bar{G}_1}\right) - w\left(\mathcal{T}^{\overrightarrow{s_1}}_{G}\right) \qquad \textbf{b)}\ w\left(\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1},q}\right) = w\left(\mathcal{T}^{\overrightarrow{s_1}}_{\bar{G}_2}\right) - w\left(\mathcal{T}^{\overrightarrow{s_1}}_{G}\right).$$

[4] $\mathcal{F}^{s_2}_{s_1,q}$, $\mathcal{F}^{s_2}_{s_1}$, $\mathcal{F}^{q}_{s_2}$ and $\mathcal{F}^{q}_{s_1}$ are equivalents to $\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1},q}$, $\mathcal{F}^{\overrightarrow{s_2}}_{\overrightarrow{s_1}}$, $\mathcal{F}^{\overrightarrow{q}}_{\overrightarrow{s_2}}$ and $\mathcal{F}^{\overrightarrow{q}}_{\overrightarrow{s_1}}$ but defined in an undirected graph.

Proof: We will only prove **a)**. Note that $w\left(\mathcal{T}_{\bar{G}_1}^{\overrightarrow{s_1}}\right) - w\left(\mathcal{T}_{G}^{\overrightarrow{s_1}}\right)$ is the weight of all the spanning trees of $\bar{G}_1$ rooted at s_1 containing the edge (s_2, s_1). It is easy to see that for any $f \in \mathcal{F}_{\overrightarrow{s_1}}^{\overrightarrow{s_2}}$, there exists a unique $t \in \mathcal{T}_{\bar{G}_1}^{\overrightarrow{s_1}}$ containing (s_2, s_1) such that $w(t) = w(f)$. Indeed, if we add the edge (s_2, s_1) to f we obtain a tree $t \in \mathcal{T}_{\bar{G}_1}^{\overrightarrow{s_1}}$ since for every node in V, there exists a unique path to s_1 (see Figure (2b)). Conversely, any tree in $t \in \mathcal{T}_{\bar{G}_1}^{\overrightarrow{s_1}}$ containing the edge (s_2, s_1) will be transformed into a forest $f \in \mathcal{F}_{\overrightarrow{s_1}}^{\overrightarrow{s_2}}$ after the removal of the edge of (s_2, s_1). Due to $\bar{w}_1\big((s_2, s_1)\big) = 1$ the equality of weights $w(f) = w(t)$ follows. The argument for **b)** is analogous to the previous case (see Figure (2c)). $\square$

The following result (Lemma 3.4) provides a closed formula for $w\left(\mathcal{F}_{\overrightarrow{s_1}}^{\overrightarrow{s_2}}\right)$ and $w\left(\mathcal{F}_{\overrightarrow{s_1},q}^{\overrightarrow{s_2}}\right)$ in terms of the in-trees of G rooted at the seeds. We defer the proof to the supplemental material.

Lemma 3.4. Let $G = (V, E, w)$ be a directed graph. Let $l_{ij}^{-1,[v]}$ represent the entry ij of the matrix $\left(L^{[v]}\right)^{-1}$ for any $v \in V$. Given s_1, s_2 and q:

$$\textbf{a)}\ w\left(\mathcal{F}_{\overrightarrow{s_1}}^{\overrightarrow{s_2}}\right) = w(\mathcal{T}^{\overrightarrow{s_1}})l_{s_2s_2}^{-1,[s_1]} = w(\mathcal{T}^{\overrightarrow{s_2}})l_{s_1s_1}^{-1,[s_2]}, \quad \textbf{b)}\ w\left(\mathcal{F}_{\overrightarrow{s_1},q}^{\overrightarrow{s_2}}\right) = w(\mathcal{T}^{\overrightarrow{s_1}})(l_{s_2s_2}^{-1,[s_1]} - l_{s_2q}^{-1,[s_1]}).$$

As a consequence of Lemma 3.4, we can conveniently find a closed form for the probabilities of the DProbWS (Definition 3):

Theorem 3.5. The Directed Probabilistic Watershed probabilities are equal to

$$\Pr(q \sim s_1) = \frac{l_{s_2s_2}^{-1,[s_1]} - l_{s_2q}^{-1,[s_1]}}{l_{s_2s_2}^{-1,[s_1]}} \text{ and } \Pr(q \sim s_2) = \frac{l_{s_2q}^{-1,[s_1]}}{l_{s_2s_2}^{-1,[s_1]}},$$

or equivalently

$$\Pr(q \sim s_2) = \frac{l_{s_1s_1}^{-1,[s_2]} - l_{s_1q}^{-1,[s_2]}}{l_{s_1s_1}^{-1,[s_2]}} \text{ and } \Pr(q \sim s_1) = \frac{l_{s_1q}^{-1,[s_2]}}{l_{s_1s_1}^{-1,[s_2]}}.$$

Our discussion was constrained to the case of two seeds only to ease our explanation. We can reduce the case of multiple seeds per label to the two seed case by merging all nodes seeded with the same label. Similarly, the case of more than two labels can be reduced to the two label scenario by using a "one versus all strategy": We choose one label and merge the seeds of other labels into one unique seed. In both cases we might add multiple edges between node pairs. While having formulated our arguments for simple graphs, they are also valid for multigraphs by the same arguments as in [12].

4 Equivalence of DProbWS and the Directed Random Walker

Resembling the ProbWS equivalence with the Random Walker, the DProbWS also turns out to be equivalent to the directed version of the Random Walker. Although multiple references exist on how to compute the absorbing probabilities of a random walker in terms of the Laplacian matrix for undirected graphs [13, 15], we could not find any suitable reference for the directed case.[5] We sketch how the absorbing probabilities can be obtained for the directed case in terms of the Laplacian matrix and refer to the supplemental material for a complete derivation.

Theorem 4.1. The probability, $x_q^{s_1}$, that a random walker on a directed graph starting at node q first reaches s_1 before reaching s_2 is given by the solution of the following linear system

$$L_U^\top x_U^{s_1} = -\left[B_1^\top\right]_{s_1}, \tag{8}$$

where $x_U^{s_i}$ represents the probability that the nodes in U are absorbed by s_i, L_U is the square submatrix of L indexed by the elements in U and $\left[B_1^\top\right]_{s_1}$ is the column s_1 of L without the entries indexed by the seeds.

[5] However, there exist closed formulas to compute the absorbing probabilities of Markov chains in terms of the fundamental matrix [17].

Proof sketch: Let P denote the transition probability matrix defined by the absorbing random walker. We note that the probability of being absorbed by s_1 at q can be obtained by computing the entry (q, s) of P^n when n tends to infinity, i.e. $x_q^{s_1} = \lim_{n\to\infty} [P^n]_{qs}$. By expressing the transition matrix in terms of the Laplacian, we find a closed formula of the matrix P^n by induction. By taking the limit to infinity, we obtain the desired result. See the supplemental material for further details. □

Remark 4.2. Note that (8) is the transposed version of the linear system solved by the undirected Random Walker and equivalently the ProbWS (3), i.e. $(L_U)\, y_U = -B_1^\top$. Since [12, 15] consider an undirected graph $\left(L_U^\top\right) = L_U$, the transposition becomes irrelevant in the undirected case.

The next theorem states the equivalence between the DProbWS and the Directed Random Walker. The proof is analogous to Theorem 4.1 in [12] and we replicate the proof in the supplemental material.

Theorem 4.3. The probability, $x_q^{s_1}$, that a random walker on a directed graph starting at node q first reaches s_1 before reaching s_2 is equal to the Directed Probabilistic Watershed's probability (Definition 3)

$$x_q^{s_1} = P(q \sim s_1).$$

5 Directed Power Watershed

In the ProbWS paper [12], it was proven that the Power Watershed [6] is equivalent to applying the ProbWS restricted to the minimum cost spanning forests. This restriction corresponds to the case of a Gibbs distribution of minimal entropy over the forests. In this section, we will prove the analogous result for the DProbWS: When the entropy of the Gibbs distribution over the directed in-forests (3.1) is minimal, then DProbWS is restricted to the minimum cost spanning in-forests (mSF). This permits us to define a natural extension of the Power Watershed to directed graphs.

In Section 3.1, we defined the weight of an edge e as $w(e) = \exp(-\mu c(e))$, where $c(e)$ was the edge-cost and μ implicitly determined the entropy of the 2-forest distribution. When $\mu \to \infty$, the distribution will have minimal entropy. Consequently, only for minimum cost spanning in-forests, mSFs (or analogously the maximum weight spanning in-forests, MSFs) the limit does not yield a zero probability, so that only they will be considered in the sampling. The proof can be found in the supplemental material.

Theorem 5.1. Given two seeds s_1 and s_2. Let further $w_{\max} := \max_{f \in \mathcal{F}_{\vec{s_1}}^{\vec{s_2}}} w(f)$. If the entropy of the Gibbs distribution over the in-forests is minimized (6), then

$$x_q^{s_1} = \frac{\left|\{f \in \mathcal{F}_{s_1,q}^{s_2} \,:\, w(f) = w_{\max}\}\right|}{\left|\{f \in \mathcal{F}_{s_1}^{s_2} \,:\, w(f) = w_{\max}\}\right|}.$$

Resembling the original Power Watershed algorithm [6], the directed version of the algorithm reduces to computing a mSF and applying the DProbWS in the plateaus (connected subgraphs with constant edge weight), resolving the ambiguity of which mSF is sampled when there is more than one.

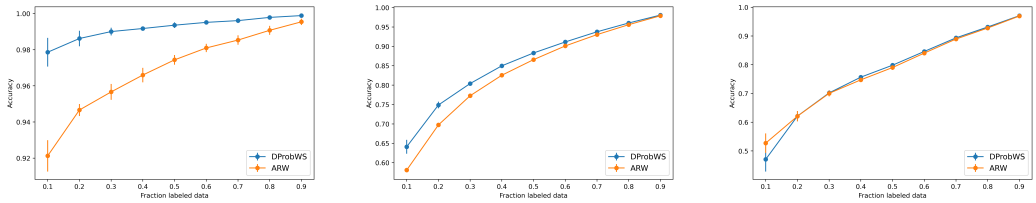
6 Experiments

Although our work has been motivated from a theoretical perspective, we run an illustrative experiment to show the performance of the DProbWS. k-Nearest Neighbors (kNN) graphs are often used in practice. Due to its asymmetric nature, they can be regarded as directed graphs. We construct kNN graphs, with $k = 5$, from the UCI datasets [10] *Digits* [6] [40] and *20Newsgroups*[7] [22]. Additionally we consider the *Email-EU*[8] network dataset from SNAP [25, 26, 42]. As a reference method, we compare our method with the one presented in [9] (referred as ARW), which also uses absorbing random walks. More details about the datasets, the ARW method and construction of the graphs can be found in the supplemental material. To evaluate the methods we use the accuracy. Inspired by [9], we sample uniformly a certain fraction r of all nodes from each class as seeds. In Figure 3, we show

[6] `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html`

[7] `https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html`

[8] `http://snap.stanford.edu/data/email-Eu-core.html`



(a) Digits (b) 20Newsgroups (c) Email-EU

Figure 3: Accuracy comparisons on the *Digits*, *20Newsgroups* and *Email-EU* datasets. In all plots, the horizontal axis denotes the fraction of labeled notes in the graph varying from 0.1 to 0.9 with 0.1 increment. We depict the average accuracy over 20 runs with error bars of one standard deviation.

the average accuracy over 20 runs for different r values between 0.1 and 0.9. In the computation of the accuracy, we also include the zero-knowledge nodes. We observe that our method obtains competitive and typically more stable results.

7 Conclusion

In this work, we presented an extension of the Probabilistic Watershed algorithm [12] that can be applied to directed graphs.

Following the rationale exposed in the Probabilistic Watershed article [12], we defined a Gibbs distribution over all the directed, seed separating, spanning in-forests rooted at the seeds of a directed graph (Definition 3). We also demonstrated, using the directed version of the Matrix Tree Theorem [24], that the Directed Probabilistic Watershed is computationally and also by result equivalent to the Directed Random Walker as in the case for the undirected version. Finally, we showed that when the entropy of the Gibbs distribution is minimized, we obtain an extension of the Power Watershed potentials [6] to directed graphs.

Fita Sanmartin et al. [12] showed that the effective resistance distance [20] triangle inequality gap of the triangle given by the two seeds and the query node is proportional to the probabilities of the Random Walker / ProbWS. In contrast, we have not been able to prove that the probabilities defined by the DProbWS are proportional to the triangle inequality gap of the effective resistance or any other distance in the directed setting. However, we conjecture that there may exist a distance / dissimilarity on the directed graph such that DProbWS probability is proportional to the triangle inequality defined by the seeds and the query node. This conjecture is left for future work.

8 Broader impact

Our work has been motivated by a theoretical understanding of already existing algorithms. The speculation of a broader impact on this kind of works is therefore less applicable. Nonetheless, we believe that the extension of the Probabilistic Watershed to directed graphs may enrich the graph-based semi-supervised learning field, where less effort has been invested in the directed framework. Furthermore, although we believe that the Directed Random Walker extension presented here must be known, we have not found any reference applying it. By delving into the relations between the spanning in-forests and the Directed Random Walker, we expect that a better understanding of the algorithm is gained and also show the community that such extension for the directed setting exists.

From a practical point of view, the application of the Directed Probabilistic Watershed / Directed Random Walker used in conjunction with other machine learning technology may be used for many applications, both good and bad. In general, any progress on semi-supervised learning will have these same consequences.

References

[1] F. Alam, S. R. Joty, and M. Imran. Graph based semi-supervised learning with convolution neural networks to classify crisis related tweets. *CoRR*, abs/1805.06289, 2018. URL `http:`

//arxiv.org/abs/1805.06289.

[2] M. Aliannejadi, M. Kiaeeha, S. Khadivi, and S. S. Ghidary. Graph-based semi-supervised conditional random fields for spoken language understanding using unaligned data. *CoRR*, abs/1701.08533, 2017. URL http://arxiv.org/abs/1701.08533.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006.

[4] P. Y. Chebotarev and E. Shamis. The matrix-forest theorem and measuring relations in small social groups1. *Automation and Remote Control*, 58:1505–1514, 1997.

[5] Y. Chong, Y. Ding, Q. Yan, and S. Pan. Graph-based semi-supervised learning: A review. *Neurocomputing*, 408:216–230, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.12.130. URL https://www.sciencedirect.com/science/article/pii/S0925231220304938.

[6] C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watershed: A unifying graph-based optimization framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[7] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1362–74, 2009. doi: 10.1109/TPAMI.2008.173.

[8] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, pages 2292–2300, 2013.

[9] J. De, X. Zhang, F. Lin, and L. Cheng. Transduction on directed graphs via absorbing random walks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1770–1784, 2018. doi: 10.1109/TPAMI.2017.2730871.

[10] D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[11] A. Erdem and M. Pelillo. Graph transduction as a non-cooperative game. In X. Jiang, M. Ferrer, and A. Torsello, editors, *Graph-Based Representations in Pattern Recognition*, pages 195–204, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-20844-7.

[12] E. Fita Sanmartin, S. Damrich, and F. A. Hamprecht. Probabilistic watershed: Sampling all spanning forests for seeded segmentation and semi-supervised learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 2780–2791. Curran Associates, Inc., 2019.

[13] F. Fouss, M. Saerens, and M. Shimbo. *Algorithms and models for network data and link analysis*. Cambridge University Press, 2016.

[14] K. Françoisse, I. Kivimäki, A. Mantrach, F. Rossi, and M. Saerens. A bag-of-paths framework for network data analysis. *Neural Networks*, 90:90–111, 2017.

[15] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006. ISSN 01628828. doi: 10.1109/TPAMI.2006.233.

[16] G. He, X. Liu, F. Fan, and J. You. Classification-aware semi-supervised domain adaptation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4147–4156, 2020. doi: 10.1109/CVPRW50498.2020.00490.

[17] J. G. Kemeny and J. L. Snell. Finite markov chains, undergraduate texts in mathematics. 1976.

[18] G. Kirchhoff. Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird. *Annalen der Physik*, 148:497–508, 1847.

[19] I. Kivimäki, M. Shimbo, and M. Saerens. Developments in the theory of randomized shortest paths with a comparison of graph node distances. *Physica A: Statistical Mechanics and its Applications*, 393:600–616, 2014.

[20] D. J. Klein and M. Randić. Resistance distance. *Journal of mathematical chemistry*, 12(1): 81–95, 1993.

[21] T. K. Koo, A. Globerson, X. Carreras, and M. Collins. Structured prediction models via the matrix-tree theorem. In *EMNLP-CoNLL*, 2007.

[22] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[23] R. Lang, R. Lu, C. Zhao, H. Qin, and G. Liu. Graph-based semi-supervised one class support vector machine for detecting abnormal lung sounds. *Applied Mathematics and Computation*, 364:124487, 2020. ISSN 0096-3003. doi: https://doi.org/10.1016/j.amc.2019.06.001. URL `https://www.sciencedirect.com/science/article/pii/S0096300319304667`.

[24] P. D. Leenheer. An elementary proof of a matrix tree theorem for directed graphs, 2019. URL `https://arxiv.org/abs/1904.12221v1`.

[25] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[26] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1):2–es, Mar. 2007. ISSN 1556-4681. doi: 10.1145/1217299.1217301. URL `https://doi.org/10.1145/1217299.1217301`.

[27] H. Linmei, T. Yang, C. Shi, H. Ji, and X. Li. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4821–4830, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1488. URL `https://www.aclweb.org/anthology/D19-1488`.

[28] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, 8(34):935–983, 2007. URL `http://jmlr.org/papers/v8/macskassy07a.html`.

[29] Q. Mei, D. Zhang, and C. Zhai. A general optimization framework for smoothing language models on graph structures. pages 611–618, 01 2008. doi: 10.1145/1390334.1390438.

[30] A. Mensch and M. Blondel. Differentiable dynamic programming for structured prediction and attention. In *ICML*, 2018.

[31] M. Senelle, S. García-Díez, A. Mantrach, M. Shimbo, M. Saerens, and F. Fouss. The sum-over-forests density index: Identifying dense regions in a graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1268–1274, 2014.

[32] Z. Song, X. Yang, Z. Xu, and I. King. Graph-based semi-supervised learning: A comprehensive review, 2021.

[33] A. Teixeira, P. Monteiro, J. Carriço, M. Ramirez, and A. Francisco. Spanning edge betweenness. In *Workshop on Mining and Learning with Graphs*, volume 24, pages 27–31, January 2013.

[34] A. Teixeira, P. Monteiro, J. Carriço, M. Ramirez, and A. P. Francisco. Not seeing the forest for the trees: Size of the minimum spanning trees (msts) forest and branch significance in mst-based phylogenetic analysis. *PLOS ONE*, 10, 2015. doi: 10.1371/journal.pone.0119315.

[35] F.-S. Tsen, T.-Y. Sung, M.-Y. Lin, L.-H. Hsu, and W. Myrvold. Finding the most vital edge with respect to the number of spanning trees. *IEEE Transactions on Reliability*, 43:600–602, 1994.

[36] W. T. Tutte. *Graph theory*. Encyclopedia of mathematics and its applications. Addison-Wesley, 1984. ISBN 0-201-13520-5 and 978-0-201-13520-6.

[37] J. E. van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Mach Learn*, 109(2): 373–440, Feb. 2020. ISSN 0885-6125, 1573-0565. doi: 10.1007/s10994-019-05855-6. URL http://link.springer.com/10.1007/s10994-019-05855-6.

[38] H. Wang, C. Ding, and H. Huang. Directed graph learning via high-order co-linkage analysis. In J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 451–466, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15939-8.

[39] G. Winkler. *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*, volume 27. Springer Science & Business Media, 2012.

[40] L. Xu, A. Krzyżak, and C. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. Syst. Man Cybern.*, 22:418–435, 1992.

[41] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han. Bridging collaborative filtering and semi-supervised learning: A neural approach for poi recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1245–1254, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098094. URL https://doi.org/10.1145/3097983.3098094.

[42] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 555–564, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098069. URL https://doi.org/10.1145/3097983.3098069.

[43] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, pages 321–328, 2004.

[44] D. Zhou, T. Hofmann, and B. Schölkopf. Semi-supervised learning on directed graphs. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1633–1640. MIT Press, 2005. URL http://papers.nips.cc/paper/2718-semi-supervised-learning-on-directed-graphs.pdf.

[45] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 1036–1043, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102482. URL https://doi.org/10.1145/1102351.1102482.

[46] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003. ISBN 1-57735-189-4. URL http://dl.acm.org/citation.cfm?id=3041838.3041953.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [No] We were unable to find the license for the datasets we used.

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]