ACTIVE SEMI-SUPERVISED LEARNING FOR DIFFUSIONS ON GRAPHS

Bishwadeep Das, Elvin Isufi and Geert Leus

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology, The Netherlands
E-mails: b.das@student.tudelft.nl; e.isufi-1@tudelft.nl; g.j.t.leus@tudelft.nl

ABSTRACT

Diffusion-based semi-supervised learning on graphs consists of diffusing labeled information of a few nodes to infer the labels on the remaining ones. The performance of these methods heavily relies on the initial labeled set, which is either generated randomly or using heuristics. The first sometimes leads to unsatisfactory results because random labeling has no guarantees to label all classes while heuristic methods only yield a good performance when multiple recursive training stages are possible. In this paper, we put forth a new paradigm for one-shot active semi-supervised learning for graph diffusions. We rephrase active learning as the problem of selecting the output labels from a label propagation model. Subsequently, we develop two methods to solve this problem and label the nodes. The first method assumes there are only a few starting labels and relies on projected compressive sensing to build the label set. The second method drops the assumption of a few starting labels and builds on sparse sensing techniques to label a few nodes. Both methods have solid mathematical grounds in signal processing and require a single training phase. Numerical results on three scenarios corroborate our findings and showcase the improved performance compared with the state of the art.

***Index Terms*—** Active learning; compressed sensing; diffusion on graphs; random walks; semi-supervised learning; sparse sensing.

1. INTRODUCTION

Learning representations for graph data is ubiquitous in social, biological, and technological networks [1]. In a social network, for instance, where users are represented by nodes and relationships by edges, a central task is to sense the network orientation on a specific topic (e.g., a new product or political orientation). Learning these representations becomes crucial in a semi-supervised setting, where acquiring labels from all nodes can be costly, time-consuming or even infeasible [2]. Label propagation —diffusing the available labels through the graph to classify the unlabeled nodes— is a method of large popularity for semi-supervised learning on graphs [3, 4, 5]. Label propagation has been recently parameterized with graph filters in [6, 7] —an approach similar to page rank and heat kernel classifiers [8, 9]— and has been further generalized with improved accuracy to class-adaptive diffusions [10]; i.e., to a classifier that learns a different graph filter for each class.

A critical aspect of diffusion-based semi-supervised classifiers is their dependence on the initial label (or training) set. This dependency gets emphasized when the number of labeled nodes is low (e.g., running a survey only on a few users in a social network), calling therefore for active semi-supervised learning methods; methods that carefully build the label set to improve the overall performance [11]. Active semi-supervised learning on graphs can be grouped in two main categories: multi-batch and single-batch training. Multi-batch methods train the classifier repeatedly to label the nodes [12, 13, 14, 15]; they start with a label set, train a classifier, label additional points, and repeat the process until a predefined metric is satisfied. Single-batch methods, as is the focus in this paper, instead avoid repetitive training and get all labels at once. Techniques within this category are proposed in [16] for Gaussian field classifiers, in [17] for graph Laplacian-based classifiers, and in [18] for graph-bandlimited data representations.

Despite the fact that diffusion methods have shown promise for semi-supervised learning on graphs, active methods for graph-diffusion learning have been little investigated. Current works in this direction treat active labeling and classification separately [15, 13, 19], i.e., the active labeling is done heuristically and these labels are then used for semi-supervised learning. In our view, this framework is more useful in a multi-batch rather than in a single-batch setting. Making active learning an integral part of the semi-supervised classifier can improve the quality of labelled nodes; hence, classification accuracy. This is especially true for the class-adaptive semi-supervised learning [10] for which framing an active learning problem is challenging.

To fill this gap, we rephrase diffusion-based active semi-supervised learning as a model output selection on graphs. Our formulation relates directly to graph diffusions and allows also to formulate and solve the active semi-supervised learning problem for class-adaptive diffusions. More concretely, our contribution is twofold: i) we postulate the problem of one-shot active diffusion-based learning on graphs —an active semi-supervised learning problem for (class-adaptive) graph diffusions— as a model output selection problem; ii) we propose two such active learning methods: one based on projected compressive sensing [20] and one based on sparse sensing [21]. Both methods pose different priors on the labeled nodes and rely on solid mathematical grounds. Numerical results on three scenarios corroborate our findings and showcase their potential for active semi-supervised learning on graphs.

The remainder of this paper proceeds as follows. Section 2 formulates the active learning problem for diffusion classifiers. Section 3 contains the proposed methods, while Section 4 the numerical results. Section 5 concludes the paper.

2. PROBLEM FORMULATION

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{1, \dots, N\}$ and edge set $\mathcal{E}$ representing the connectivity between nodes. The graph structure is represented through the graph shift operator matrix $\mathbf{S}$; an $N \times N$ symmetric matrix in which the (i, j)th entry $[\mathbf{S}]_{ij}$ is nonzero only if $(i, j) \in \mathcal{E}$ or if $i = j$. Typical examples for $\mathbf{S}$ are the graph adjacency matrix $\mathbf{A}$, the graph Laplacian

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ with $\mathbf{D}$ the degree matrix or any of their normalized or translated forms. One such form is $\mathbf{S} = \mathbf{A}\mathbf{D}^{-1}$ used to model a random walk on graphs. A random walk of length one (or one hop) can be regraded as a discrete-time Markov chain with each node being a state and in which the transition probability of landing at node j from node i is $\Pr\{j|i\} = [\mathbf{S}]_{ji}$; a random walk of length K is a sequence of K random hops.

Random walks are used for semi-supervised learning on graphs through label propagation [10, 8, 22, 23] . The goal is to classify nodes among C candidate classes by having labeled only a few of them. Specifically, let $\mathcal{V}_c \subset \mathcal{V}$ be the subset of nodes labeled to class $c = 1, \ldots, C$. A random walk starts from these nodes with starting probability $\mathbf{p}_c^{(0)} = [p_{1c}^{(0)}, \ldots, p_{Nc}^{(0)}]^\top \in \mathbb{R}^N$ in which the ith entry for class c

$$p_{ic}^{(0)} = \begin{cases} \frac{1}{|\mathcal{V}_c|} & \text{if } i \in \mathcal{V}_c, \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

denotes the probability of starting the walk at node i. The starting probabilities are uniform within each class c and $|\mathcal{V}_c|$ is the number of nodes labeled in class c. Since the shift operator matrix $\mathbf{S}$ respects the structure of the graph, the one-hop landing probability vector for class c can be written as $\mathbf{p}_c^{(1)} = \mathbf{S}\mathbf{p}_c^{(0)}$, where the ith entry $p_{ic}^{(1)}$ is the probability of landing at node i having started from $\mathbf{p}_c^{(0)}$. Likewise, the landing probability vector for class c for a walk of length K is $\mathbf{p}_c^{(K)} = \mathbf{S}\mathbf{p}_c^{(K-1)} = \mathbf{S}^K\mathbf{p}_c^{(0)}$. A graph-based diffusion classifier combines the probabilities $\mathbf{p}_c^{(0)}, \mathbf{p}_c^{(1)}, \ldots, \mathbf{p}_c^{(K)}$ with a class-specific vector of coefficients $\mathbf{h}_c = [h_{c0}, \ldots, h_{cK}]^\top$ to obtain the final diffusion probabilities

$$\mathbf{q}_c(\mathbf{h}_c) = \sum_{k=0}^{K} h_{ck}\mathbf{p}_c^{(k)} = \sum_{k=0}^{K} h_{ck}\mathbf{S}^k\mathbf{p}_c^{(0)} \tag{2}$$

for class c. For future reference, let us define the diffusion filter for class c as

$$\mathbf{H}_c(\mathbf{S}) = \sum_{k=0}^{K} h_{ck}\mathbf{S}^k \tag{3}$$

and write (2) as $\mathbf{q}_c(\mathbf{h}_c) = \mathbf{H}_c(\mathbf{S})\mathbf{p}_c^{(0)}$.

The parameters $\mathbf{h}_c$ are estimated to match a target probability vector $\bar{\mathbf{q}}_c$ with ith entry

$$\bar{q}_{ic} = \begin{cases} \frac{1}{|\bar{\mathcal{V}}|} & \text{if } i \in \mathcal{V}_c, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\bar{\mathcal{V}} = \cup_{c=1}^{C}\mathcal{V}_c$ is the set of all labeled nodes with $|\bar{\mathcal{V}}| = M$. Put simply, the diffusion parameters of class c, $\mathbf{h}_c$, are obtained by equating the ith entry of (2) to (4) yet only for the labeled nodes in $\bar{\mathcal{V}}$. To avoid overfitting, the estimation of these parameters is regularized with graph-priors on the diffused probabilities $\mathbf{q}_c(\mathbf{h}_c)$ in (2), e.g., smoothness. This boils down to solving the optimization problem

$$\begin{aligned} \underset{\mathbf{h}_c}{\text{minimize}} \quad & \mathcal{L}(\bar{\mathbf{q}}_c, \mathbf{q}_c(\mathbf{h}_c)) + \gamma\mathcal{R}(\mathbf{q}_c(\mathbf{h}_c), \mathbf{S}) \\ \text{subject to} \quad & \mathbf{h}_c \succeq \mathbf{0},\ \mathbf{h}_c^T\mathbf{1} = 1. \end{aligned} \tag{5}$$

where $\mathcal{L}(\bar{\mathbf{q}}_c, \mathbf{q}_c(\mathbf{h}_c))$ is a distance measure between the target value $\bar{\mathbf{q}}_c$ and the diffused probabilities $\mathbf{q}_c(\mathbf{h}_c)$ calculated only over the labeled nodes $\bar{\mathcal{V}}$ while $\mathcal{R}(\mathbf{q}_c(\mathbf{h}_c), \mathbf{S})$ is the graph-based regularizer for the diffused probabilities. The two constraints ensure that the estimated parameters $\mathbf{h}_c$ yield an output in (2) that is a probability mass function for class c.

Given then $\mathbf{q}_1(\mathbf{h}_1), \ldots, \mathbf{q}_C(\mathbf{h}_C)$, the unlabeled nodes $i \in \mathcal{V}\backslash\bar{\mathcal{V}}$ are assigned to the class that yields

$$\underset{c\in\{1,\ldots,C\}}{\text{argmax}} \quad q_{ic}(\mathbf{h}_c) \quad \text{for } i = 1, \ldots, |\mathcal{V}\backslash\bar{\mathcal{V}}| \tag{6}$$

where $q_{ic}(\mathbf{h}_c)$ is the ith entry of $\mathbf{q}_c(\mathbf{h}_c)$ [10].

While (2) regards class-adaptive parameters $\mathbf{h}_c$, two other popular approaches consider the same parameters $\mathbf{h}_c = \mathbf{h}\ \forall c$: the personalized page rank classifier fixes $\mathbf{h} = (1-h)[h^0, h^1, \ldots, h^K]^\top$ with scalar $0 \leq h \leq 1$ [8]; the heat kernel classifier fixes $\mathbf{h} = e^{-h}[1, h, \frac{h^2}{2}, \ldots, \frac{h^K}{K!}]^\top$ with scalar $h \geq 0$ [9]. In this work, we will leverage both the class-adaptive and non-adaptive scenarios.

As it follows from (2), the set of labeled nodes $\bar{\mathcal{V}}$ (i.e., $\mathbf{p}_c^{(0)}$ in (1) and $\bar{\mathbf{q}}_c$ in (4)) plays an important role in diffused semi-supervised learning. In specific, the location of these nodes w.r.t. the graph topology influences the diffusion output $\mathbf{q}_c(\mathbf{h}_c)$ in (2), and hence, the estimated parameters in (5) as well as the classifier output in (6); all these quantities depend on the starting nodes of the walk, i.e., $\mathbf{p}_c^{(0)}$. Random labeling does not account for the graph structure and the diffusion process on top of it, leading to unrepresentative nodes and low classification accuracy. This is particularly true for one-shot or single batch active learning. In this work, we tackle this issue under the aforementioned one-shot scenario and build the labeled set $\bar{\mathcal{V}}$ with solid mathematical tools to improve the classification accuracy for adaptive graph-based diffusion classifiers. This problem, which we label as *active diffusion learning on graphs* is formalized as follows.

Problem statement. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose nodes can be classified into C classes with the diffusion process in (2) from the labeled nodes $\bar{\mathcal{V}} \subset \mathcal{V}$; the task is to build the label set $\bar{\mathcal{V}}$ from scratch in a one-shot setting as the diffusion starting nodes with per-class probabilities given by (1).

3. ACTIVE LEARNING

We formulate the active learning problem as designing an $M \times N$ sampling matrix $\mathbf{C}$ to select the $M < N$ entries of $\mathbf{q}_c(\mathbf{h}_c)$ in (2) that carry the most information about the starting probabilities $\mathbf{p}_c^{(0)}$. Formally, matrix $\mathbf{C}$ belongs to the combinatorial set

$$\mathcal{C}_{M,N} = \{\mathbf{C} \in \{0,1\}^{M\times N} : \mathbf{C}\mathbf{1}_N = \mathbf{1}_M, \mathbf{C}^\top\mathbf{1}_N \preceq \mathbf{1}_N\} \tag{7}$$

that selects M out of N different nodes and satisfies $\mathbf{C}\mathbf{C}^\top = \mathbf{I}_M$ and $\mathbf{C}^\top\mathbf{C} = \text{diag}(\mathbf{c})$, where $\mathbf{1}_M$ is the $M \times 1$ vector of all ones, $\mathbf{I}_M$ is the $M \times M$ identity matrix, and $\mathbf{c} \in \{0,1\}^N$ is an $N \times 1$ vector with $c_i = 1$ if and only if node i is labeled, i.e., belongs to $\bar{\mathcal{V}}$.

With this in place, we write the diffusion classifier output for class c on the selected nodes as

$$\tilde{\mathbf{q}}_c(\mathbf{h}_c) = \mathbf{C}\mathbf{q}_c(\mathbf{h}_c) = \mathbf{C}\mathbf{H}_c(\mathbf{S})\mathbf{p}_c^{(0)}. \tag{8}$$

Remark that during active learning we do not know the labeled set $\bar{\mathcal{V}}$; hence, the target probability vector $\bar{\mathbf{q}}_c$ (4), which further implies that we cannot estimate a class-specific parameter vector $\mathbf{h}_c$ as per (5). To tackle this issue, we follow a two-step approach. First, we consider a known and fixed parameter vector $\mathbf{h} = \mathbf{h}_c\ \forall c$ (e.g., the personalized page rank parameters) to build the label set $\bar{\mathcal{V}}$ during active learning. Then, we follow the class-adaptive approach in (1)-(6)

with the set $\bar{\mathcal{V}}$ previously built to label the remaining nodes. Thus, during active learning, equation (8) becomes

$$\tilde{\mathbf{q}}_c(\mathbf{h}) = \mathbf{C}\mathbf{q}_c(\mathbf{h}) = \mathbf{C}\mathbf{H}(\mathbf{S})\mathbf{p}_c^{(0)}. \tag{9}$$

That is, the role of $\mathbf{C}$ is now that of selecting the M rows of the known and fixed diffusion filter $\mathbf{H}(\mathbf{S})$ that best describes the diffusion of $\mathbf{p}_c^{(0)}$ over the graph.

We develop two methods for building $\mathbf{C}$ (i.e., $\bar{\mathcal{V}}$). The first method interprets $\mathbf{p}_c^{(0)}$ as a sparse vector and relies on compressed sensing to select M rows of $\mathbf{H}(\mathbf{S})$ that are closer to an equiangular frame [24]. The second method drops the sparsity assumption and leverages sparse sensing to select the M rows of $\mathbf{H}(\mathbf{S})$ that lead to the minimum volume of the confidence ellipsoid [21].

3.1. Compressed sensing active learning

Problems of the form in (8) with a sparse $\mathbf{p}_c^{(0)}$ have been widely studied in compressed sensing literature and fall under the category of optimized projections for sparse recovery [20, 24, 25, 26]. These works design a general (not binary) $M \times N$ projection matrix $\mathbf{C}$ such that the resulting matrix $\mathbf{C}\mathbf{H}(\mathbf{S})$ is close to an equiangular frame [27]; that is, close to an $M \times N$ dictionary matrix $\mathbf{E}$ in which the inner products of any two columns are equal in absolute value. The inner products of all columns of $\mathbf{E}$ can be obtained through the Gram matrix $\mathbf{G}_e = \mathbf{E}^\top\mathbf{E}$, which has entries $[\mathbf{G}_e]_{ij}$ of absolute value

$$\big|[\mathbf{G}_e]_{ij}\big| = \begin{cases} \sqrt{\frac{N-M}{M(N-1)}} & i \neq j \\ 1 & i = j \end{cases}. \tag{10}$$

Our goal is, therefore, to design a sampling matrix $\mathbf{C}$ such that the resulting Gram matrix

$$\mathbf{G}_c = \mathbf{H}^\top(\mathbf{S})\mathbf{C}^\top\mathbf{C}\mathbf{H}(\mathbf{S}) = \mathbf{H}^\top(\mathbf{S})\text{diag}(\mathbf{c})\mathbf{H}(\mathbf{S}) \tag{11}$$

has entries $[\mathbf{G}_c]_{ij}$ with absolute value close to (10). But since the (i, j)th entry of $\mathbf{H}(\mathbf{S})$ satisfies $[\mathbf{H}(\mathbf{S})]_{ij} \geq 0$ by construction — recall $\mathbf{q}_c(\mathbf{h})$ should be a probability vector; see also (5)— the entries of $\mathbf{G}_c$ are all nonnegative. It is thus sufficient to show that $[\mathbf{G}_c]_{ij}$ itself (without absolute value) is close to (10). The identity matrix $\mathbf{I}_N$ is another example of $\mathbf{G}_e$ and can also be used to design projection matrices [25]. We now pose the design of $\mathbf{C}$ as solving the optimization problem

$$\begin{aligned} \underset{\mathbf{c}}{\text{minimize}} \quad & ||\mathbf{H}^\top(\mathbf{S})\text{diag}\big(\mathbf{c}\big)\mathbf{H}(\mathbf{S}) - \mathbf{G}_e||_F^2 \\ \text{subject to} \quad & ||\mathbf{c}||_0 = M, \quad \mathbf{c} \in \{0,1\}^N \end{aligned} \tag{12}$$

where the cost function measures with the Frobenius norm $\|\cdot\|_F$ the distance between the sampled Gram matrix $\mathbf{G}_c$ in (11) and the equiangular frame Gram matrix $\mathbf{G}_e$ in (10). The optimization constraints ensure the resulting matrix $\mathbf{C}$ is a selection matrix. Problem (12) is a combinatorial NP-hard problem. We can solve it efficiently by substituting the l_0 pseudo-norm $\|\mathbf{c}\|_0 = M$ with the l_1 norm surrogate $\|\mathbf{c}\|_1 = M$ and the Boolean constraint $\mathbf{c} \in \{0,1\}^N$ with the box one $\mathbf{c} \in [0,1]^N$; the latter transform (12) into a convex problem. Relaxing the problem leads often to solutions that are far from the optimal one. We have found instead that solving (12) with greedy methods, i.e., starting with the set $\bar{\mathcal{V}} = \mathcal{V}$ and removing one node at a time that decreases the cost the least until $|\bar{\mathcal{V}}| = M$, leads often to better results. As far as we know, it has not been proven to be sub-modular.

A few remarks are now in order. First, an equiangular frame is not guaranteed to exist for any tuple (M, N) [27]; in general, M has to be larger than a specific value that depends on N. In these cases, even solving the original problem (12) may not give rise to a good label set $\bar{\mathcal{V}}$. Second, differently from [20, 24], we avoid the repeated projections since $\mathbf{H}(\mathbf{S})$ is known in our case and also the projection matrix $\mathbf{C}$ has a well-defined binary structure. Third, if we resort to the convex approach, we can also regularize the solution of (12) with a term $\mathcal{R}(\mathbf{H}(\mathbf{S}), \mathbf{c}, \mathbf{S})$ on how the selected labels diffuse over the graph; e.g., $\mathcal{R}(\mathbf{H}(\mathbf{S}), \mathbf{c}, \mathbf{S}) = \mathbf{c}^\top\mathbf{H}^\top(\mathbf{S})\mathbf{L}\mathbf{H}(\mathbf{S})\mathbf{c}$ imposes that the diffused labels on the nodes in $\mathbf{c}$ are smooth over the graph. We have seen that this improves the performance of the convex approach but still is slightly worse than greedy.

3.2. Sparse sensing active learning

The compressed sensing active learning (CS-AL) relies on the fact that $\mathbf{p}_c^{(0)}$ is sparse. However, we can also construct the labeled set $\bar{\mathcal{V}}$ without this assumption by relying on a sparse sensing framework [21]. In sparse sensing active learning (SS-AL), we drop the sparsity assumption and assume the true labels $\mathbf{q}_c^{\text{true}}$ for class c can be written as the final diffused probabilities up to some uncertainty

$$\mathbf{q}_c^{\text{true}} = \mathbf{q}_c(\mathbf{h}) + \mathbf{n} = \mathbf{H}(\mathbf{S})\mathbf{p}_c^{(0)} + \mathbf{n} \tag{13}$$

where $\mathbf{q}_c(\mathbf{h})$ is the model landing probability vector for class c and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_M)$. Since estimating $\mathbf{p}_c^{(0)}$ is linked to $\mathbf{q}_c^{\text{true}}$ through the pseudo-inverse of $\mathbf{C}\mathbf{H}(\mathbf{S})$, SS-AL selects the M rows of $\mathbf{H}(\mathbf{S})$ that yield the minimum estimation error.

Denoting with $\mathbf{h}_i(\mathbf{S}) \in \mathbb{R}^N$ the ith row of $\mathbf{H}(\mathbf{S})$ we can write the ith entry of $\mathbf{q}_c^{\text{true}}$ in (13) as

$$q_{ic}^{\text{true}} = \mathbf{h}_i^\top(\mathbf{S})\mathbf{p}_c^{(0)} + n_i. \tag{14}$$

where n_i is the ith entry of $\mathbf{n}$. Selecting the M nodes to label implies selecting the M rows of $\mathbf{H}(\mathbf{S})$ that lead to the minimum estimation error on the starting probability vector $\mathbf{p}_c^{(0)}$. Among the different choices to measure the estimation error, we consider the log-determinant of the error covariance matrix. This metric relates to the volume of the confidence ellipsoid and captures the uncertainty about the estimate of $\mathbf{p}_c^{(0)}$ [21]. Selecting the M nodes to label then implies solving the combinatorial problem

$$\begin{aligned} \underset{\mathbf{c}}{\text{minimize}} \quad & \text{logdet}\big(\mathbf{H}^\top(\mathbf{S})\text{diag}(\mathbf{c})\mathbf{H}(\mathbf{S}) + \epsilon\mathbf{I}_N\big) \\ \text{subject to} \quad & ||\mathbf{c}||_0 = M, \quad \mathbf{c} \in \{0,1\}^N \end{aligned} \tag{15}$$

where $\epsilon\mathbf{I}_N$ ensures the existence of the log-determinant. The benefit of the log-determinant over alternative cost functions is that it is is sub-modular. As such, it allows to avoid relaxation techniques and build $\bar{\mathcal{V}}$ with greedy methods. Algorithm 1 provides the greedy solution for (15). Since the term $\big(\mathbf{H}^\top(\mathbf{S})\text{diag}(\mathbf{c})\mathbf{H}(\mathbf{S})\big)$ is always rank deficient, we should select those nodes that increase the condition number of the non-singular part the most; hence, the term $\epsilon\mathbf{I}_N$. It should be noted that the two proposed approaches do not take the classification accuracy into consideration while building $\bar{\mathcal{V}}$.

4. NUMERICAL RESULTS

We considered three node classification scenarios, namely a stochastic block model (SBM), a random sensor network (RSN), and a Facebook subnetwork [28]. During active learning, we considered the parameter vector $\mathbf{h}$ to be that of the personalized page rank with

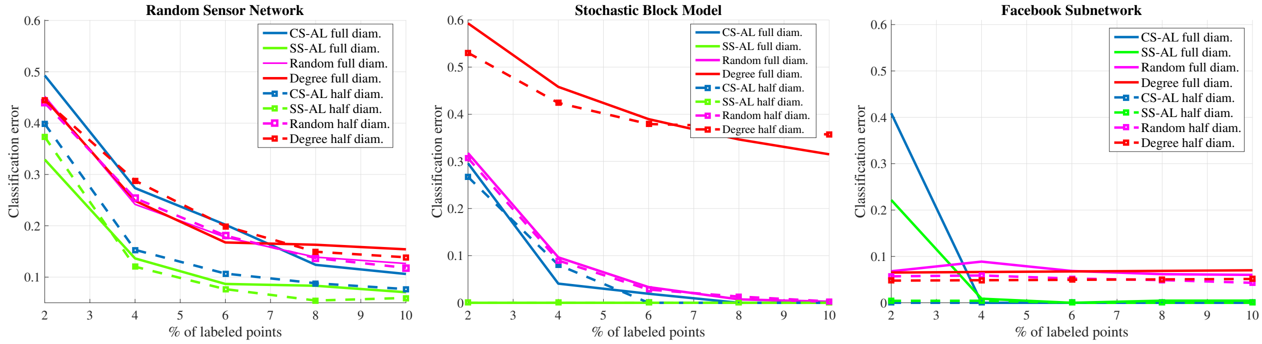

Fig. 1. Classification error versus percentage of labeled nodes for the proposed CS-AL and SS-AL and for the random and degree-based active learning. The results are shown for two filter orders K in (3): K being the graph diameter and K being half of the graph diameter. Both proposed methods improve w.r.t random and degree-based labeling, where the SS-AL reaches also zero classification error in well-clustered scenarios (e.g., SBM graph and Facebook subnetwork). For CS-AL in the RSN and SBM the target matrix is $\mathbf{G}_e$ (10), while for the Facebook subnetwork the target matrix is the identity matrix.

	Class 1				Class 2			
	CS-AL	SS-AL	Random	Degree	CS-AL	SS-AL	Random	Degree
Class 1	214	214	208.6	213	0	0	5	4
Class 2	0	0	7.5	11	14	14	6.9	0

Table 1. Confusion matrix for the proposed CS-AL and SS-AL, random labeling, and degree-based labeling on the Facebook subnetwork for $|\bar{\mathcal{V}}| = 6$ and filter order $K = 4$. Each row shows how the different algorithms classify the nodes belonging to that class.

Algorithm 1 Greedy solution for problem (15)

```
Set the cardinality of labeled set $|\bar{\mathcal{V}}| = M$; the global parameters
  $\mathbf{h}$ for all classes in (3); $\bar{\mathcal{V}} = \emptyset$; $m = 0$
while $m \leq M$ do
    Select the node $j$ that
    $j = \underset{j \in \mathcal{V}\setminus\bar{\mathcal{V}}}{\text{argmax}} \ \text{logdet}\big(\sum_{i\in\bar{\mathcal{V}}} \frac{1}{\sigma_i^2}\mathbf{h}_i\mathbf{h}_i^T + \frac{1}{\sigma_j^2}\mathbf{h}_j\mathbf{h}_j^\top\big)$
    $\bar{\mathcal{V}} = \bar{\mathcal{V}} \cup j$
    $m = m + 1$
end while
```

$h = 0.9$ [8]. We analyzed the diffusion filters in (3) with two different orders K: first, K is the graph diameter and second the half of it. The proposed CS-AL (Section 3.1) and SS-AL (Section 3.2) methods are compared with random labeling whose results are averaged over 100 realizations and with degree-based heuristic labeling (i.e., label the M nodes with the largest degree).

The SBM and the RSN have both $N = 200$ nodes to be classified into $C = 4$ classes. The SBM has 4 blocks, average diameter 4, and intra- and inter-block probabilities of 0.8 and 0.01, respectively. The RSN is constructed with the default settings in the GSP toolbox [29] and has average diameter 15. All results for SBM and RSN are averaged over ten different graph realizations. The Facebook subnetwork has $N = 234$ nodes clustered in two connected and non-balanced communities of 219 and 15 nodes and diameter 8. The goal is to label the most relevant users for classifying through adaptive diffusions to which of the $C = 2$ communities the remaining users belong to.

Fig. 1 shows the classification error for different cardinalities of the labeled set $\bar{\mathcal{V}}$ expressed as percentages w.r.t. the total number of nodes. Overall, the proposed methods improve the classification accuracy of random labeling: for scenarios with a more distinctive clustering behavior (i.e., SBM and Facebook subgraph) the SS-AL achieves zero classification error. The CS-AL falls back in performance for low values of M (i.e., $|\bar{\mathcal{V}}|$); this is because the equiangular frame conditions are violated. But when these conditions hold (i.e., larger M) the CS-AL reaches also optimal performance. We also see that increasing K from half to the full graph diameter does not lead to any improvement and it might also degrade the performance (see Facebook subnetwork). This is because a larger K accumulates at each node labeled information also from the nodes in the other classes; hence, degrading the overall performance. Therefore, as it is good practice in diffusion semi-supervised learning, also for active semi-supervised learning it is beneficial to account only for label propagation in the vicinity of a node (e.g., low K).

Table 3.2 shows the confusion matrix for the Facebook subnetwork. The cell (i, j) denotes the number of nodes belonging to class i and classified to class j. These results confirm those in Fig. 1, i.e., the proposed methods outperform the other alternatives. As such, we conclude that model-driven active learning has a large potential to improve semi-supervised learning on graphs since it accounts for both the network topology and the diffusion process on top of it.

5. CONCLUSION

We proposed a one-shot active semi-supervised learning on graphs for diffusion-based classifiers. The proposed solution rephrased the active learning problem as the problem of output label selection in a label propagation model. We then developed two active learning methods: the first method relies on compressed sensing; the second method leverages sparse sensing methods. Numerical tests on three scenarios showed the proposed approaches improve over random and heuristic degree-based labeling. In the near future, we will investigate the connection between the selected nodes and the graph spectral representation of the diffusion filter.

6. REFERENCES

[1] M. Newman, *Networks*, Oxford university press, 2018.

[2] X. Zhu, "Semi-supervised learning literature survey," Tech. Rep., University of Wisconsin-Madison Department of Computer Sciences, 2005.

[3] X. Zhou and Z.Ghahramani, "Learning from labeled and unlabeled data with label propagation," .

[4] Y. Bengio, O. Delalleau, and N. Le Roux, "Label propagation and quadratic criterion," 2006.

[5] W. Liu, J. Wang, and S. Chang, "Robust and scalable graph-based semisupervised learning," *Proceedings of the IEEE*, vol. 100, no. 9, pp. 2624–2638, 2012.

[6] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[7] S. Chen, F. Cerda, P. Rizzo, J. Bielak, J. H. Garrett, and J. Kovačević, "Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring," *IEEE Transactions on Signal Processing*, vol. 62, no. 11, pp. 2879–2893, 2014.

[8] F. Lin and W. W. Cohen, "Semi-supervised classification of network data using very few labels," in *2010 International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 2010, pp. 192–199.

[9] K. Kloster and D. F. Gleich, "Heat kernel based community detection," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1386–1395.

[10] D. Berberidis, A. N. Nikolakopoulos, and G. B. Giannakis, "Adaptive diffusions for scalable learning over graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 5, pp. 1307–1321, 2018.

[11] Burr Settles, "Active learning literature survey," Tech. Rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.

[12] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani, "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," in *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, 2003, vol. 3, pp. 58–65.

[13] J. Long, J. Yin, W. Zhao, and E. Zhu, "Graph-based active learning based on label propagation," in *International Conference on Modeling Decisions for Artificial Intelligence*. Springer, 2008, pp. 179–190.

[14] M. Bilgic, L. Mihalkova, and L. Getoor, "Active learning for networked data," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 79–86.

[15] L. Shi, Y. Zhao, and J. Tang, "Batch mode active learning for networked data," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 2, pp. 33, 2012.

[16] M. Ji and J. Han, "A variance minimization criterion to active learning on graphs," in *Artificial Intelligence and Statistics*, 2012, pp. 556–564.

[17] Q. Gu, T. Zhang, J. Han, and C. H. Ding, "Selective labeling via error bound minimization," in *Advances in neural information processing systems*, 2012, pp. 323–331.

[18] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 492–501.

[19] L. Mingwei, Y. Yukai, C. Jianjun, L. Weiming, and C. Xiaoyun, "Active semi-supervised community detection algorithm with label propagation," in *International Conference on Database Systems for Advanced Applications*, 2013.

[20] M. Elad, "Optimized projections for compressed sensing," *IEEE Transactions on Signal Processing*, vol. 55, no. 12, pp. 5695–5702, 2007.

[21] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2008.

[22] I. M. Kloumann, J. Ugander, and J. Kleinberg, "Block models and personalized pagerank," *Proceedings of the National Academy of Sciences*, vol. 114, no. 1, pp. 33–38, 2017.

[23] E. Merkurjev, A. L. Bertozzi, and F. Chung, "A semi-supervised heat kernel pagerank mbo algorithm for data classification," *Communications in Mathematical Sciences*, vol. 16, no. 5, pp. 1241–1265.

[24] J. A. Tropp, I. S. Dhillon, R. W. Heath, and T. Strohmer, "Designing structured tight frames via an alternating projection method," *IEEE Transactions on information theory*, vol. 51, no. 1, pp. 188–209, 2005.

[25] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1395–1408, 2009.

[26] G. Li, Z. Zhu, D. Yang, L. Chang, and H. Bai, "On projection matrix optimization for compressive sensing systems," *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2887–2898, 2013.

[27] P. G. Casazza, D. Redmond, and J. C. Tremain, "Real equiangular frames," in *2008 42nd annual conference on information sciences and systems*. IEEE, 2008, pp. 715–720.

[28] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Advances in neural information processing systems*, 2012, pp. 539–547.

[29] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "Gspbox: A toolbox for signal processing on graphs," *arXiv preprint arXiv:1408.5781*, 2014.