RANDOM WALKS WITH RESTARTS FOR GRAPH-BASED CLASSIFICATION: TELEPORTATION TUNING AND SAMPLING DESIGN

Dimitris Berberidis[1,2], *Athanasios N. Nikolakopoulos*[2] *and Georgios B. Giannakis*[1,2]

[1]Dept. of ECE and [2]Digital Tech. Center, University of Minnesota
Minneapolis, MN 55455, USA
Emails: {bermp001,anikolak,georgios}@umn.edu

ABSTRACT

The present work introduces methods for sampling and inference for the purpose of semi-supervised classification over the nodes of a graph. The graph may be given or constructed using similarity measures among nodal features. Leveraging the graph for classification builds on the premise that relation among nodes can be modeled via stationary distributions of a certain class of random walks. The proposed classifier builds on existing scalable random-walk-based methods and improves accuracy and robustness by automatically adjusting a set of parameters to the graph and label distribution at hand. Furthermore, a sampling strategy tailored to random-walk-based classifiers is introduced. Numerical tests on benchmark synthetic and real labeled graphs demonstrate the performance of the proposed sampling and inference methods in terms of classification accuracy.

Index Terms— Random Walks, Sampling on Graphs, PageRank, Semi-Supervised Learning, Seed Set Expansion

1. INTRODUCTION

In many machine learning tasks, data are not available in a feature-based representation; instead, only a set of relations between instances can be leveraged for inference. In bipartite networks for example nodes may correspond to users and items, for which no information is available other that the interactions between them [1]. Although networks may arise naturally in certain applications (e.g. social interactions, transportation infrastructure, biological dependencies) they can in general be constructed from any set of nodal feature vectors using proper similarity measures; see e.g., [2, 3]. In either case, the task of semi-supervised learning (SSL) for classification over graphs boils down to observing a subset of the nodes, and then making predictions on the remaining unobserved ones. SSL on graphs has been approached by a variety of intertwined methods based on the principles of iterative label propagation [4, pp. 193-216],[5, 6, 7], while also viewed under the scope of kernels on graphs [8], random walks [9, 10], graph partitioning [11], and transductive learning [12].

In this context, the present work aims at improving the performance of random-walk-based classifiers. The latter are of particular interest in the presence of large graphs since they enjoy high scalability by fully leveraging graph sparsity. The proposed classifier builds on an efficient parametrization of random walks; parameters are then tuned in order to increase classification confidence on nodes with known labels aiming at boosting accuracy on unlabeled nodes. The proposed tuning is solved via quadratic programming at the dimension of available samples which are typically very few. Thus, the computational overhead over standard (un-tuned) random-walks is relatively small. Subsequently, the desire for classifiers with high confidence over the entire graph and for any label distribution inspired the development of a novel sampling strategy aimed at increasing the accuracy of random-walk-based classifiers. The rest of the paper is organized as follows. Section 2 formally introduces SSL random-walk-based classification on graphs. Our proposed tuning method is presented in Section 3, while the novel sampling scheme is introduced in Section 4. Finally, numerical experiments are presented in Section 5.

2. MODELING AND PROBLEM STATEMENT

Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}$ is the set of N nodes and $\mathcal{E}$ the set of edges. In general, a weight matrix $\mathbf{W}$ is also present with $W_{i,j} > 0$ if $(v_i, v_j) \in \mathcal{E}$. Furthermore, consider that a discrete label $y_i \in \mathcal{Y}$ corresponds to each node v_i. In SSL for classification over graphs, a subset $\mathcal{L} \subset \mathcal{V}$ of nodes is assigned labels $\mathbf{y}_{\mathcal{L}}$, and the goal is inferring the labels of the remaining nodes belonging to the unlabeled set $\mathcal{U} = \mathcal{V} \setminus \mathcal{L}$. Given a measure of influence, a node that is influenced more by labeled nodes of a certain class is assumed to also belong to the same class. Thus, *label-propagation* on graphs boils down to quantifying the influence of $\mathcal{L}$ on $\mathcal{U}$, see, e.g. [7, 10, 13]. A simple, intuitive and easy to compute measure of nodal influence can be obtained via the concept of

Work in this paper was supported by the NSF Grants No. 1500713, No. 1514056, and No. 1711471

random walks on graphs.

The simple random walk on a graph is essentially a Markov chain where the state space is the set of nodes, and the transition probabilities are given as $\Pr\{X_t = v_i | X_{t-1} = v_j\} = W_{i,j}/d_j$, where X_t denotes the position of the random walker (state) at slot t, $d_j = \sum_{k \in \mathcal{N}_j} W_{k,j}$ is the degree of vertex (or node) v_j and $\mathcal{N}_j$ its neighborhood [14, 15]. The stationary distribution of the X_t is given by the dominant right eigenvector of the *column stochastic* transition probability matrix $\mathbf{H} := \mathbf{W}\mathbf{D}^{-1}$, where $\mathbf{D} = \operatorname{diag}(d_1, d_2, \ldots, d_N)$ [16]. Consider now the Random Walk with Restart (RwR) (also known as personalized Pagerank [17]) that is further parametrized by a teleportation vector $\boldsymbol{\theta}$ and a teleportation probability d, with transition probabilities

$$\Pr\{X_t = v_i | X_{t-1} = v_j; \boldsymbol{\theta}, d\} = \left\{ \begin{array}{ll} W_{i,j}/d_j, & w.p.\ 1-d \\ \theta_i, & w.p.\ d \end{array} \right. . \tag{1}$$

Unlike the simple random walk where transitions occur only to neighboring nodes, each transition in the RwR is accompanied by a "cointoss" that with probability d may "teleport" the random walk onto other nodes of the graph. It follows readily that the transition probability matrix of the RwR in (1) is expressed as $\mathbf{P} = (1-d)\mathbf{H} + d\boldsymbol{\theta}\mathbf{1}^T$, where $\mathbf{1}$ is the $N-$length vector containing ones. Interestingly, by setting $\boldsymbol{\theta} = \mathbf{e}_i$ where $\mathbf{e}_i$ is the $i-$th canonical vector, the random walk teleports back only to the $i-$th node. Such nodes are often referred to as "seeds"; see, e.g. [9]. The key idea is that the steady-state distribution $\boldsymbol{\pi}$ encapsulates the influence of the seed v_i on the entire graph. Intuitively speaking, the closer any node v_j is to v_i in the random-walk sense, the larger the value π_j is expected to be. In general, multiple seeds belonging to $\mathcal{S} \subset \mathcal{V}$ collectively influence the graph by seeding the RwR via non-zero values in the corresponding entries of the teleportation vector, that is $\theta_i > 0$, $\forall i \in \mathcal{S}$ and $\theta_i = 0$, $\forall i \notin \mathcal{S}$. In a classification-over-the-graph scenario, a stationary distribution $\boldsymbol{\pi}(k)$ can be computed for each class $k \in \mathcal{Y}$ where the seeding set is $\mathcal{L}_k = \{i \in \mathcal{L} : y_i = k\}$, i.e. labeled nodes that belong to class k. If $\mathcal{L}_k = \emptyset$ for some class, one may set $\boldsymbol{\pi}(k) = \mathbf{0}$ for simplicity. Upon obtaining $\{\boldsymbol{\pi}(k)\}_{k \in \mathcal{Y}}$, label predictions can be made over the graph as $\hat{y}_i = \arg\max_{k \in \mathcal{Y}} \boldsymbol{\pi}_i(k)$, $\forall i \in \mathcal{U}$ that is, by selecting the class with corresponding labeled nodes exhibiting maximum influence.

Existing random-walk-based label propagation schemes typically do not discriminate among labeled nodes; see, e.g.,[8, 5, 9]; each of them enjoys equal seeding contribution to their corresponding $\boldsymbol{\pi}(k)$ via *uniform* per-class teleportation distributions $\boldsymbol{\theta}_k = \boldsymbol{\theta}_y^{\text{Unif}}$, where

$$[\boldsymbol{\theta}_k^{\text{Unif}}]_i = \left\{ \begin{array}{ll} \frac{1}{|\mathcal{L}_k|}, & i \in \mathcal{L}_k \\ 0, & \text{otherwise} \end{array} \right. .$$

However, in principle, higher classification accuracy should be achievable for (unknown) non-uniform $\boldsymbol{\theta}_k$'s. The ensuing section proposes a scalable method for improving RwR's by tuning $\boldsymbol{\theta}_k$'s.

Algorithm 1 Tuned RwR for SSL on graphs

```
Input: $\mathbf{W}, d, \mathcal{L}, \mathbf{y}_\mathcal{L}$
Compute $\mathbf{G}_{:,\mathcal{L}}$ as in (3) via batch Power Method
Obtain $\boldsymbol{\theta}^*_{\text{LS}}$ by solving (5)
Predict unlabeled nodes as $\hat{\mathbf{y}}_\mathcal{U} = \operatorname{sign}(\mathbf{G}_{\mathcal{U},\mathcal{L}}\boldsymbol{\theta}^*)$
```

3. TUNED RANDOM WALKS

The present section introduces methods for improving the accuracy of RwR-based classifiers by tuning the per-class teleportation distributions $\boldsymbol{\theta}_k$. The main idea is to parametrize the per-class stationary distributions $\boldsymbol{\pi}(k)$ that act as soft labels by the corresponding $\boldsymbol{\theta}_k$ as $\boldsymbol{\pi}(k; \boldsymbol{\theta}_k)$. Subsequently, we leverage the fact that $\boldsymbol{\pi}(k; \boldsymbol{\theta}_k)$ which is formally obtained via the power method (see, e.g., [9]) is also given as

$$\boldsymbol{\pi}(k; \boldsymbol{\theta}_k) = \mathbf{G}_{:,\mathcal{L}_k}\boldsymbol{\theta}_k \ \forall k \in \mathcal{Y}, \tag{2}$$

where $\mathbf{G}_{:,\mathcal{L}_k}$ contains the subset of columns of matrix $\mathbf{G} = (\mathbf{I} - (1-d)\mathbf{H})^{-1}$ that correspond to $\mathcal{L}_k$. Overall, the per-node stationary distributions that correspond to $\mathcal{L} = \cup_{k \in \mathcal{Y}} \mathcal{L}_k$ suffice for SSL classification and can be obtained in batch form as columns of $N \times |\mathcal{L}|$ matrix $\mathbf{G}_{:,\mathcal{L}}$ which satisfies the following matrix equation

$$\mathbf{G}_{:,\mathcal{L}} = (1-d)\mathbf{H}\mathbf{G}_{:,\mathcal{L}} + d\mathbf{S}_\mathcal{L} \tag{3}$$

and can be computed via batch power method; here, $\mathbf{S}_\mathcal{L}$ is the $N \times |\mathcal{L}|$ selection matrix such that $\mathbf{S}_\mathcal{L} = [\ldots, \mathbf{e}_i, \ldots] \ \forall i \in \mathcal{L}$. While the complexity of performing (3) is $\mathcal{O}(|\mathcal{L}|N^2)$, exploiting the sparsity of $\mathbf{H}$ reduces the former to just $\mathcal{O}(|\mathcal{L}||\mathcal{E}|)$. Parametrization of d remains a challenging direction that is being pursued outside the scope of this paper. In our context, d is typically small enough to boosts accuracy via far-reaching random walks while being large enough to ensure convergence of (3) in a few iterations (cf. [18]). Having parametrized per-class RwRs (cf. (2)), let us denote the length $|\mathcal{L}|$ vector $\boldsymbol{\theta}$ implicitly containing the non-zero elements of $\boldsymbol{\theta}_k$ at entries that correspond to $\mathcal{L}_k$ and allowing for the formulation of the following general problem

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}} \sum_{i \in \mathcal{L}} f(\hat{y}_i, y_i; \boldsymbol{\theta}) + \lambda g(\boldsymbol{\theta}) \\ &\text{s.t. } \mathbf{e}_{\mathcal{L}_k}^T \boldsymbol{\theta} = 1 \ \forall k \in \mathcal{Y}, \ \ \mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{1} \end{aligned} \tag{4}$$

where $f(\hat{y}_i, y_i; \boldsymbol{\theta})$ is the fitting loss, $g(\boldsymbol{\theta})$ is a regularization term, $\mathbf{e}_\mathcal{S}$ denotes a vector of 1's at entries indicated by $\mathcal{S}$ and 0's elsewhere, and the simplex constrains ensure that the resulting $\{\boldsymbol{\theta}_k^*\}_{k \in \mathcal{Y}}$ are valid probability mass functions. The aim is to tune $\boldsymbol{\theta}$ in order to increase the confidence (or decrease the fitting loss) with which the RwR-based classifier predicts the nodes whose true labels are known. While various type of costs (e.g. hinge loss, Huber loss) may act as $f(\cdot)$, a simple choice is the quadratic loss that for $\mathcal{Y} = \{0, 1\}$ is

$$f(\hat{y}_i, y_i; \boldsymbol{\theta}_{-1}, \boldsymbol{\theta}_1) = (\boldsymbol{\pi}_i(1; \boldsymbol{\theta}_1) - \boldsymbol{\pi}_i(-1; \boldsymbol{\theta}_{-1}) - y_i)^2,$$

with the general optimization problem in (4) becoming the following quadratic program

$$\boldsymbol{\theta}^*_{LS} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{G}_{\mathcal{L},\mathcal{L}}\mathbf{D}_y\boldsymbol{\theta} - \mathbf{y}_{\mathcal{L}}\|_2^2 + \lambda\|\boldsymbol{\theta} - \boldsymbol{\theta}^{\text{Unif}}\|_2^2$$
$$\text{s.t. } \mathbf{e}^T_{\mathcal{L}_{-1}}\boldsymbol{\theta} = 1,\ \mathbf{e}^T_{\mathcal{L}_1}\boldsymbol{\theta} = 1,\ \mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{1} \quad (5)$$

where $\mathbf{D}_y = \text{diag}(\mathbf{y}_{\mathcal{L}})$, and the regularizer penalizes large deviations from non-informative uniform teleportation distributions encapsulated in $\boldsymbol{\theta}^{\text{Unif}}$. Obtained $\boldsymbol{\theta}^*_{LS}$ is then used to yield predictions $\hat{\mathbf{y}}_{\mathcal{U}}$ on the remaining unlabeled. Overall, the proposed Tuned RwR method is summarized as Algorithm 1. In terms of complexity, $\mathcal{O}(|\mathcal{L}||\mathcal{E}|)$ is required for (3), $\mathcal{O}(|\mathcal{L}|^3)$ for (5) and $\mathcal{O}(|\mathcal{L}|N)$ for the predictions. Since typically $|\mathcal{L}| \ll N$, the complexity of the method scales linearly with N allowing for application to large-scale graphs. Regarding the case of multiple classes, one may follow the one-vs-the-rest approach and solve (5) for each class in $\mathcal{Y}$ with all other classes treated as -1.

4. MINIMUM COHERENCE SAMPLING

The Tuned RwR presented in the previous section is readily applicable for classification when the graph and set of labeled nodes is given. Nevertheless, given a budget of k labels and the freedom to design $\mathcal{L}$ with $|\mathcal{L}| \leq k$, one may further improve classification accuracy by judiciously selecting which nodes to sample. For instance, [19] and [20] developed sampling designs for GMRF-based SSL classification on graphs (see also our recent work [21]), while labeling nodes with high Pagerank scores was proposed in [9] for RwR-based classifiers. Towards this goal, the present section puts forth a sample design method that is tailored to RwR-based classifiers.

In Tuned RwR's (cf. (5)) the aim is at increasing the confidence with which the classifier predicts the known labels. The notion can be generalized to describe the prediction confidence over the entire graph for any label configuration over a given (candidate) label set $\mathcal{L}$. Ideally, in the binary case, one would prefer classifiers that yield large margins $\{|\pi_i(1) - \pi_i(-1)|\}_{i\in\mathcal{V}}$ between the predicted soft labels. Thus, for a given $\mathcal{L}$, finding a relatively large $\epsilon > 0$ such that

$$\|\boldsymbol{\pi}(1;\boldsymbol{\theta}_1) - \boldsymbol{\pi}(-1;\boldsymbol{\theta}_{-1})\|_2^2 \geq \epsilon, \quad (6)$$

holds forall $\{\mathcal{L}_1, \mathcal{L}_{-1} \subseteq \mathcal{V} : \mathcal{L}_1 \cup \mathcal{L}_{-1} = \mathcal{L}\}$ and $\boldsymbol{\theta}_1, \boldsymbol{\theta}_{-1}$ on the simplex, implies that all possible classifiers based on (Tuned) RwR's are sufficiently confident. Note that the left-hand term in (6) can be written as $\|\boldsymbol{\pi}(1;\boldsymbol{\theta}_1)\|_2^2 + \|\boldsymbol{\pi}(-1;\boldsymbol{\theta}_{-1})\|_2^2 - 2\boldsymbol{\pi}^T(1;\boldsymbol{\theta}_1)\boldsymbol{\pi}(-1;\boldsymbol{\theta}_{-1})$ where

$$\|\boldsymbol{\pi}(1;\boldsymbol{\theta}_1)\|_2^2 = \sum_{i\in\mathcal{L}_1}\|\mathbf{g}_i\|_2^2[\boldsymbol{\theta}_1]_i^2 + 2\sum_{\substack{i,j\in\mathcal{L}_1\\ j\neq i}}\mathbf{g}_i^T\mathbf{g}_j[\boldsymbol{\theta}_1]_i[\boldsymbol{\theta}_1]_j$$
$$\geq \sum_{i\in\mathcal{L}_1}\|\mathbf{g}_i\|_2^2[\boldsymbol{\theta}_1]_i^2 \geq \frac{1}{|\mathcal{L}_1|}\min_{i\in\mathcal{L}_1}\|\mathbf{g}_i\|_2^2 \quad (7)$$

with $\mathbf{g}_i$ denoting the i^{th} column of $\mathbf{G}_{:,\mathcal{L}}$ (cf, (2)). The first inequality follows from the positivity of all quantities involved,

Algorithm 2 Greedy approximation of MiCo set of nodes

```
Input: $\mathbf{W}, d, k$
Compute $\mathbf{G}$ as in (3) with $\mathcal{L} = \mathcal{V}$ via batch Power Method
Initialize: $\mathcal{L}^{(0)} = \arg\max_{i\in\mathcal{V}}\|\mathbf{g}_i\|_2,\ \mathcal{U}^{(0)} = \mathcal{V}/\mathcal{L}^{(0)}$
for $t = 1 : k-1$ do
    Obtain $\{c_j^{\mathcal{L}^{(t-1)}}\}_{j\in\mathcal{U}^{(t-1)}}$ as in (10)
    $i^* = \arg\min_{j\in\mathcal{U}^{(t-1)}} c_j^{\mathcal{L}^{(t-1)}}$
    $\mathcal{L}^{(t)} = \mathcal{L}^{(t-1)} \cup \{i^*\},\ \mathcal{U}^{(t)} = \mathcal{U}^{(t-1)}/\{i^*\}$
end for
Return set of indices $\mathcal{L}^{(k)}$
```

and the second inequality from the fact that $\boldsymbol{\theta}_1$ lies on the probability simplex; similarly for $\|\boldsymbol{\pi}(-1;\boldsymbol{\theta}_{-1})\|_2^2$. Furthermore, the cross-products arising from (6) are bounded as

$$\boldsymbol{\pi}^T(1;\boldsymbol{\theta}_1)\boldsymbol{\pi}(-1;\boldsymbol{\theta}_{-1}) = \sum_{i\in\mathcal{L}_1}\sum_{j\in\mathcal{L}_{-1}}\mathbf{g}_i^T\mathbf{g}_j[\boldsymbol{\theta}_1]_i[\boldsymbol{\theta}_{-1}]_j$$
$$\leq \sum_{i\in\mathcal{L}_1}\sum_{j\in\mathcal{L}_{-1}}\mathbf{g}_i^T\mathbf{g}_j$$
$$\leq |\mathcal{L}_1||\mathcal{L}_{-1}|\max_{i\in\mathcal{L}_1}\max_{j\in\mathcal{L}_{-1}}\mathbf{g}_i^T\mathbf{g}_j$$
$$\leq \frac{|\mathcal{L}|^2}{4}\max_{i,j\in\mathcal{L}}\mathbf{g}_i^T\mathbf{g}_j \quad (8)$$

A closer look at (7) and (8) reveals that finding $\mathcal{L}$ that satisfies (6) with the largest ϵ can be approximated by maximizing $\min_{i\in\mathcal{L}}\|\mathbf{g}_i\|_2$ while minimizing $\max_{i,j\in\mathcal{L}}\mathbf{g}_j^T\mathbf{g}_i$. The latter is closely tied to solving

$$\mathcal{L}^* = \arg\min_{\mathcal{L}\subset\mathcal{V}:|\mathcal{L}|=k} C(\mathbf{G}_{:,\mathcal{L}}) \quad (9)$$

where
$$C(\mathbf{G}_{:,\mathcal{L}}) = \max_{\substack{i,j\in\mathcal{L}\\ i\neq j}}\frac{\mathbf{g}_i^T\mathbf{g}_j}{\|\mathbf{g}_i\|_2\|\mathbf{g}_j\|_2}$$

is the maximum angle between any pair of columns of $\mathbf{G}_{:,\mathcal{L}}$, also termed as matrix *coherence* [22]. Solving (9) in order to obtain the minimum coherence (MiCo) set of nodes $\mathcal{L}^*$ incurs exponential complexity since all $k-$sized subsets of $\mathcal{V}$ need to be evaluated. In practice, a MiCo set of size k can be approximated using a greedy iterative algorithm that expands a given MiCo set $\mathcal{L}^{(t)}$ at iteration t to $\mathcal{L}^{(t+1)} = \mathcal{L}^{(t)} \cup i^*$, where i^* is the index of the node $i \in \mathcal{U}^{(t)} = \mathcal{V} \setminus \mathcal{L}^{(t)}$ with corresponding $\mathbf{g}_i$ that has *maximum angle* with the linear subspace formed by the columns of $\mathbf{G}_{:,\mathcal{L}^{(t)}}$. Equivalently, one may select $i \in \mathcal{U}^{(t)}$ that yields minimum normalized inner product coefficient

$$c_i^{\mathcal{L}^{(t)}} = \frac{\mathbf{g}_i^T\mathbf{p}_i^{\mathcal{L}^{(t)}}}{\|\mathbf{g}_i\|_2\|\mathbf{p}_i^{\mathcal{L}^{(t)}}\|_2}, \quad (10)$$

where $\mathbf{p}_i^{\mathcal{L}^{(t)}} = \mathbf{P}_{\mathcal{L}^{(t)}}\mathbf{g}_i$ is the orthogonal projection of $\mathbf{g}_i$ onto the linear subspace spanned by the columns of $\mathbf{G}_{:,\mathcal{L}^{(t)}}$, with

$$\mathbf{P}_{\mathcal{L}^{(t)}} = \mathbf{G}_{:,\mathcal{L}^{(t)}}(\mathbf{G}^T_{:,\mathcal{L}^{(t)}}\mathbf{G}_{:,\mathcal{L}^{(t)}})^{-1}\mathbf{G}^T_{:,\mathcal{L}^{(t)}}$$

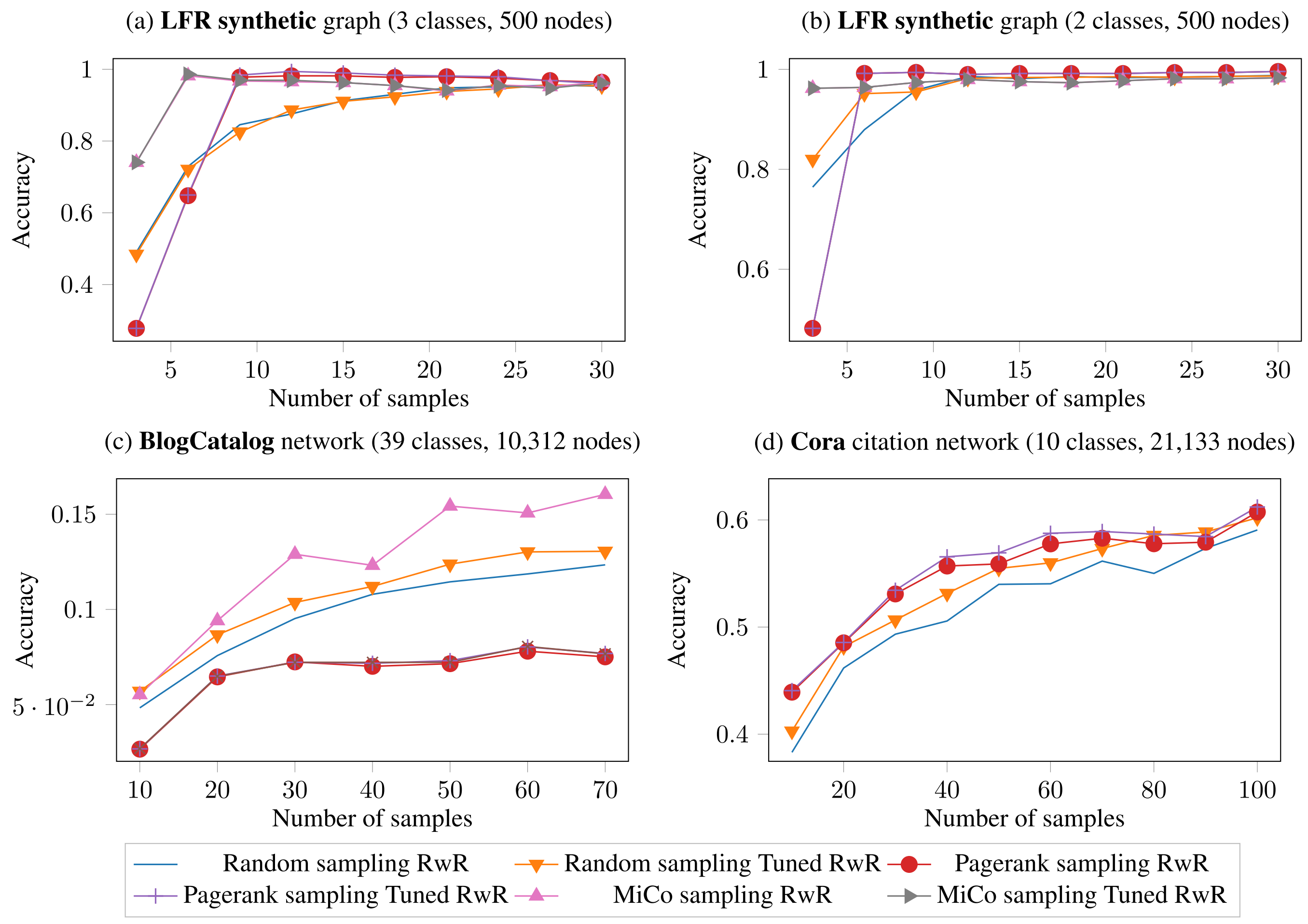

Fig. 1. Accuracy vs number of labeled nodes

being the corresponding projection operator [23] which can be updated recursively as new columns are added. The proposed method for approximating a MiCo set of k nodes is summarized as Algorithm 2 and incurs $\mathcal{O}(k^2N^2)$ complexity which can be affordable for small to medium-size graphs. For larger graphs, one may consider preselecting a subset $\mathcal{L}_{\text{pre}} \subseteq \mathcal{V}$ and then applying Algorithm 2 to $\mathcal{L}_{\text{pre}}$ with $\mathcal{O}(k^2|\mathcal{L}_{\text{pre}}|^2)$ complexity.

5. NUMERICAL TESTS

Experiments on synthetic and real-world labeled graphs where run in order to assess the performance of the proposed Tuned RwR classifiers as well as the MiCo sampling strategy. Fig.1 depicts classification accuracy (weighted F1-score) on benchmark graphs as a function of the number of labels that were obtained. All random walks were performed with $d = 0.05$ which yields a good tradeoff between accuracy and complexity (cf. (3)), and with $\lambda = 0.3$; experiments involving random samples where averaged over 30 Monte Carlo runs. Two synthetic graphs generated according to the LFR benchmark method [24] were tested first, one with 3 classes (Fig.1(a)) and one with 2 (Fig.1(b)). The resulting plots indicate that the proposed tuning methods improve the classification accuracy of RwRs, especially when samples are picked at random. Furthermore, the proposed MiCo sampling strategy significantly outperforms Pagerank [9] when very few samples are afforded. As expected, MiCo sampling produces extremely confident classifiers that offer little room for improvement via tuning. As is well documented, graph structures and label distributions may vary greatly among real world networks. Our results on the BlogCatalog network [25] depicted in Fig.1(c) show Pagerank sampling (even combined with Tuned RwRs) to be highly inaccurate, significantly less fit that random sampling. On the other hand, the proposed MiCo sampling enjoys superior accuracy throughout the range of afforded labels. Finally, plotted in Fig.1(d) are results on the large Cora citation network available in [26]. MiCo sampling was not implemented for this graph due to memory limitations. Each node of Cora corresponds to a scientific paper belonging to one (or more) fields used as class labels for our experiments; for nodes with more than one classes only the first was used. Again it was observed that tuning improves the performance of RwRs with the improvement being marginal for Pagerank sampling but significant for random sampling.

6. REFERENCES

[1] A. N. Nikolakopoulos, A. Korba, and J. D. Garofalakis, “Random surfing on multipartite graphs,” in *2016 IEEE International Conference on Big Data (Big Data)*, Dec 2016, pp. 736–745.

[2] G. V. Karanikolas, G. B. Giannakis, K. Slavakis, and R. M. Leahy, “Multi-kernel based nonlinear models for connectivity identification of brain networks,” in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Proc.*, Shanghai, China, March 2016.

[3] B. B. Y. Shen and G. B. Giannakis, “Kernel-based structural equation models for topology identification of directed networks,” *IEEE Trans. Sig. Proc.*, vol. 65, no. 10, pp. 2503–2516, May 2017.

[4] Y. Bengio, O. Delalleau, and N. Le Roux, *Label propagation and quadratic criterion*. MIT Press, January 2006.

[5] X. Zhu, Z. Ghahramani, and J. Lafferty, “Semi-supervised learning using Gaussian fields and harmonic functions,” in *Proc. of Intl. Conf. on Machine Learning*, Washington DC, Aug. 2003.

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

[7] B. Kveton, M. Valko, A. Rahimi, and L. Huang, “Semi-supervised learning with max-margin graph cuts,” in *Proc. of. Intl. Conf. on Artif. Intell. and Stat.*, 2010, pp. 421–428.

[8] E. Merkurjev, A. L. Bertozzi, and F. Chung, “A semi-supervised heat kernel pagerank MBO algorithm for data classification,” 2016.

[9] F. Lin and W. W. Cohen, “Semi-supervised classification of network data using very few labels,” in *Proc. of Intl. Conf. on Advances in Social Net. Analysis and Mining*. IEEE, 2010, pp. 192–199.

[10] X.-M. Wu, Z. Li, A. M. So, J. Wright, and S.-F. Chang, “Learning with partially absorbing random walks,” in *Proc. of Adv. in Neural Inform. Proc. Systems*, Harrahs and Harveys, Lake Tahoe, Dec. 2012, pp. 3077–3085.

[11] J. Ugander and L. Backstrom, “Balanced label propagation for partitioning massive graphs,” in *Proc. of the Intl. Conf. on Web Search and Data Mining*, 2013, pp. 507–516.

[12] P. P. Talukdar and K. Crammer, “New regularized algorithms for transductive learning,” in *Proc. of. the Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, 2009, pp. 442–457.

[13] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-supervised learning,” *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[14] S. Brin and L. Page, “Reprint of: The anatomy of a large-scale hypertextual web search engine,” *Computer networks*, vol. 56, no. 18, pp. 3825–3833, 2012.

[15] A. N. Nikolakopoulos and J. D. Garofalakis, “Ncdawarerank: A novel ranking method that exploits the decomposable structure of the web,” in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, ser. WSDM ’13. New York, NY, USA: ACM, 2013, pp. 143–152.

[16] R. Serfozo, *Basics of applied stochastic processes*. Springer Science & Business Media, 2009.

[17] H. Tong, C. Faloutsos, and J.-Y. Pan, “Random walk with restart: fast solutions and applications,” *Knowledge and Information Systems*, vol. 14, no. 3, pp. 327–346, Mar 2008.

[18] T. Haveliwala and S. Kamvar, “The second eigenvalue of the google matrix,” Stanford, Tech. Rep., 2003.

[19] Y. Ma, R. Garnett, and J. Schneider, “σ-optimality for active learning on Gaussian random fields,” in *Proc. of Adv. in Neural Inf. Proc. Systems*, Lake Tahoe, Dec. 2013.

[20] M. Ji and J. Han, “A variance minimization criterion to active learning on graphs.” in *Intl. Conf. on Artif. Intel. and Stat.*, La Palma, Canary Islands, April 2012.

[21] D. Berberidis and G. B. Giannakis, “Data-adaptive active sampling for efficient graph-cognizant classification,” *ArXiv e-prints. [Online]. Available: arXiv:1705.07220v2.*, 2017.

[22] J. A. Tropp, “Just relax: convex programming methods for identifying sparse signals in noise,” *IEEE Trans. Info. Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[23] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012.

[24] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” *Physical review E*, vol. 78, no. 4, pp. 046–110, 2008.

[25] R. Zafarani and H. Liu, “Social computing data repository at ASU,” 2009. [Online]. Available: http://socialcomputing.asu.edu

[26] J. Kunegis, “Konect: The koblenz network collection,” in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW ’13 Companion. New York, NY, USA: ACM, 2013, pp. 1343–1350.