Scalable Learning with Privacy over Graphs

Shen, Yanning; Leus, Geert

DOI
10.1109/DSW.2019.8755782

Publication date
2019

Document Version
Final published version

Published in
2019 IEEE Data Science Workshop, DSW 2019 - Proceedings

SCALABLE LEARNING WITH PRIVACY OVER GRAPHS

Yanning Shen[*]*, and Geert Leus*[†]

[*]Dept. of ECE and DTC, University of Minnesota, Minneapolis, USA
[†]Dept. of ECE, Delft University of Technology, Delft, The Netherlands

ABSTRACT

Graphs have well-documented merits for modeling complex systems, including financial, biological, and social networks. Network nodes can also include attributes such as age or gender of users in a social network. However, the size of real-world networks can be massive, and nodal attributes can be unavailable. Moreover, new nodes may emerge over time, and their attributes must be inferred in real time. In this context, the present paper deals with scalable learning of nodal attributes by estimating a nodal function based on noisy observations at a subset of nodes. A multikernel-based approach is developed which is scalable to large-size networks. The novel method is capable of providing real-time evaluation of the function values on newly-joining nodes without resorting to a batch solver. In addition, the novel scheme only relies on an encrypted version of each node's connectivity, which promotes privacy. Experiments on real datasets corroborate the effectiveness of the proposed methods.

1. INTRODUCTION

Graphs and networks emerge in various areas, such as social, brain, and power networks. Functions of nodes can represent certain attributes or classes of these nodes. In Facebook for instance, each node represents a person, and the presence of an edge indicates that two persons are friends, while nodal attributes can be age, gender or movie ratings of each person.

However, there are often unknown nodal function values, due to, e.g., privacy issues. Hence, a topic of great practical importance is to interpolate missing nodal values (class, ranking or function), based on the function values at a subset of observed nodes. Function estimation over graphs based on partial observations has been investigated extensively, [1, 2, 3, 4, 5, 6, 7]. It has also been studied recently as signal reconstruction over graphs, see e.g., [8, 9, 10, 11, 12], where signal values on unobserved nodes can be estimated by properly introducing a graph-aware prior. Kernel-based methods for learning over graphs offer a unifying framework that includes linear and nonlinear function estimators [10, 13, 14]. The nonlinear methods outperform the linear ones but suffer from the curse of dimensionality [15], rendering them less attractive for large-scale networks. To alleviate this limitation, a scalable kernel-based approach will be introduced in the present paper, which leverages the random feature (RF) approximation to ensure *scalability* while also allowing *real-time* evaluation of the functions over large-scale dynamic networks.

In certain applications, new nodes may join the network over time, which requires real-time evaluation of the nodal function values. Existing rigorous approaches are in general less efficient in accounting for newly-joining nodes, and need to solve the problem over all nodes in the network, every time new nodes join the network. To this end, this paper develops a scalable *online graph-adaptive* algorithm that can efficiently infer nodal functions even on newly-joining nodes 'on the fly.'

Besides scalability and adaptivity, nodes may have high *privacy* requirements, therefore may not be willing to reveal their connectivities. Most graph-based learning methods however, require knowing the entire network adjacency, and thus cannot meet the privacy requirements. Our novel RF-based approach on the other hand, only requires an encrypted version of each node's connectivity pattern, which makes it appealing for networks with stringent privacy constraints.

2. KERNEL-BASED LEARNING OVER GRAPHS

Consider a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ of N nodes, whose topology is captured by a known adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$. Let $a_{nn'} \in \mathbb{R}$ denote the (n, n') entry of $\mathbf{A}$, which is nonzero only if an edge is present from node n' to n. A real-valued function (or signal) on a graph is a mapping $f : \mathcal{V} \to \mathbb{R}$, where $\mathcal{V}$ is the set of vertices. The value $f(v) = x_v$ represents an attribute of $v \in \mathcal{V}$, e.g., in a social network, x_{v_n} could denote the age of the nth person. Suppose that a collection of noisy samples $\{y_m = x_{v_{n_m}} + e_m\}_{m=1}^M$ is available, where e_m models noise, and $M \leq N$ represents the number of measurements. Given $\{y_m\}_{m=1}^M$, and with the graph topology known, the goal is to estimate $f(v)$, and thus reconstruct the graph signal at unobserved vertices. Letting $\mathbf{y} := [y_1, \ldots, y_M]^\top$, the observation vector obeys $\mathbf{y} = \mathbf{\Psi}\mathbf{x} + \mathbf{e}$, where $\mathbf{x} := [x_{v_1}, \ldots, x_{v_N}]^\top$, $\mathbf{e} := [e_1, \ldots, e_M]^\top$, and $\mathbf{\Psi} \in \{0, 1\}^{M\times N}$ is a sampling matrix with binary entries $[\mathbf{\Psi}]_{m,n_m} = 1$, and 0, elsewhere.

Given $\mathbf{\Psi}$, $\mathbf{y}$, and $\mathbf{A}$, the goal is to estimate $\mathbf{x}$ over the entire network. Consider function f belonging to a reproducing kernel Hilbert space (RKHS) defined as [13, 10] $\mathcal{H} := \{f :$

This work was supported by NSF 1500713 and NSF 1711471.

$f(v) = \sum_{n=1}^{N} \alpha_n \kappa(v, v_n), \alpha_n \in \mathbb{R}\}$, where $\kappa : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is a pre-selected kernel function. Hereafter, we will let $n_m = m$ for notational convenience, and without loss of generality (wlog). Given $\mathbf{y}$, the RKHS-based estimate is formed as

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{M} \sum_{m=1}^{M} \mathcal{C}(f(v_m), y_m) + \mu \Omega \left(\|f\|_{\mathcal{H}}^2 \right) \quad (1)$$

where the cost $\mathcal{C}(\cdot, \cdot)$ can be selected depending on the learning task, e.g., the least-squares (LS) for regression, or the logistic loss for classification; $\|f\|_{\mathcal{H}}^2 := \sum_n \sum_{n'} \alpha_n \alpha_{n'} \kappa(v_n, v_{n'})$ is the RKHS norm; $\Omega(\cdot)$ is an increasing function; and, $\mu > 0$ is a regularization parameter that copes with overfitting.

According to the representer theorem, the optimal solution of (1) admits the finite-dimensional form given by $\hat{f}(v) = \sum_{m=1}^{M} \alpha_m \kappa(v, v_m) := \boldsymbol{\alpha}^\top \mathbf{k}(v)$ [13, 10], where $\boldsymbol{\alpha} := [\alpha_1 \dots \alpha_M]^\top$ and $\mathbf{k}(v) := [\kappa(v, v_1) \dots \kappa(v, v_M)]^\top$. This implies that the function over the graph can be estimated by optimizing over the $M \times 1$ vector $\boldsymbol{\alpha}$ [cf. (1)]

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \frac{1}{M} \sum_{m=1}^{M} \mathcal{C}(\boldsymbol{\alpha}^\top \mathbf{k}(v_m), y_m) + \mu \Omega \left(\boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \right) \quad (2)$$

where $\mathbf{K} := \boldsymbol{\Psi}^\top \bar{\mathbf{K}} \boldsymbol{\Psi}$, and the $N \times N$ kernel matrix $\bar{\mathbf{K}}$ has entries $[\bar{\mathbf{K}}]_{n,n'} := \kappa(v_n, v_{n'})$. Graph-kernel based approaches were developed in [10, 13], where given the normalized Laplacian matrix $\mathbf{L} := \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} := \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top$, with $\mathbf{D} := \mathrm{diag}(\mathbf{A}\mathbf{1})$, the family of graphical kernels is

$$\bar{\mathbf{K}} := r^\dagger(\mathbf{L}) := \mathbf{U} r^\dagger(\boldsymbol{\Lambda}) \mathbf{U}^\top \quad (3)$$

with $r(.)$ a non-decreasing scalar function of the eigenvalues. By selecting $r(.)$, different graph properties can be accounted for, including smoothness, band-limitedness, the random walk [13], and diffusion [2].

It can be observed from (3) that formulating $\bar{\mathbf{K}}$ generally requires eigenvalue decomposition of $\mathbf{L}$, which incurs complexity $\mathcal{O}(N^3)$ that can be prohibitive for large-scale networks. Moreover, the graph-kernel-based scheme requires knowledge of the topology, meaning $\mathbf{A}$, in order to estimate the nodal function of each node. In response to these challenges, an online scalable kernel-based method will be developed in the present paper to deal with sequentially obtained data samples, over generally dynamic networks.

3. GRAPH-ADAPTIVE LEARNING OVER GRAPHS

3.1. RF-based learning over graphs

Instead of resorting to a graph kernel that requires an eigenvalue decomposition of $\mathbf{L}$ in (3), the present section advocates treating the *connectivity pattern of each node as its feature vector*, which can be the nth column $\mathbf{a}_n^{(c)}$ and possibly the nth row $(\mathbf{a}_n^{(r)})^\top$ of the adjacency. We will henceforth term this *connectivity pattern* of v_n, and denote it as $\mathbf{a}_n$, for brevity. Given $\mathbf{a}_n$, we will interpolate unavailable nodal function values $\hat{f}(v_n)$ using a nonparametric approach, that is different and scalable relative to [13] and [10]. The kernel matrix is now $[\bar{\mathbf{K}}]_{n,n'} = \kappa(v_n, v_{n'}) = \kappa(\mathbf{a}_n, \mathbf{a}_{n'})$. Again, with M nodes sampled, the representer theorem asserts that the sought function estimator has the form [15]

$$\hat{f}(v_n) = \hat{f}(\mathbf{a}_n) = \sum_{m=1}^{M} \alpha_m \kappa(\mathbf{a}_m, \mathbf{a}_n) := \boldsymbol{\alpha}^\top \mathbf{k}(\mathbf{a}_n) \quad (4)$$

where $\mathbf{k}(\mathbf{a}_n) := [\kappa(\mathbf{a}_n, \mathbf{a}_1) \dots \kappa(\mathbf{a}_n, \mathbf{a}_M)]^\top$.

To bypass the growing complexity due to the computation of the kernel matrix, we will resort to the so-called random feature approximation [16] in order to reduce the original functional learning task in (2) to a finite space with the number of unknown parameters not growing with M. We first approximate κ using random features (RFs) [16, 17, 18] that are obtained from a shift-invariant kernel satisfying $\kappa(\mathbf{a}_n, \mathbf{a}_{n'}) = \kappa(\mathbf{a}_n - \mathbf{a}_{n'})$. For $\kappa(\mathbf{a}_n - \mathbf{a}_{n'})$ absolutely integrable, its Fourier transform $\pi_\kappa(\mathbf{v})$ exists and represents the power spectral density, which upon normalizing to ensure $\kappa(\mathbf{0}) = 1$, can also be viewed as a probability density function (pdf); hence, $\kappa(\mathbf{a}_n - \mathbf{a}_{n'}) = \int \pi_\kappa(\mathbf{v}) e^{j\mathbf{v}^\top(\mathbf{a}_n - \mathbf{a}_{n'})} d\mathbf{v} := \mathbb{E}_{\mathbf{v}} \left[e^{j\mathbf{v}^\top(\mathbf{a}_n - \mathbf{a}_{n'})} \right]$, where the last equality is due to the definition of the expected value. Drawing D independent and identically distributed samples $\{\mathbf{v}_i\}_{i=1}^D$ from $\pi_\kappa(\mathbf{v})$, the ensemble mean can be approximated by the sample average

$$\hat{\kappa}(\mathbf{a}_n, \mathbf{a}_{n'}) = \mathbf{z}_{\mathbf{V}}^\top(\mathbf{a}_n) \mathbf{z}_{\mathbf{V}}(\mathbf{a}_{n'}) \quad (5)$$

where $\mathbf{V} := [\mathbf{v}_1, \dots, \mathbf{v}_D]^\top \in \mathbb{R}^{D \times N}$, and $\mathbf{z}_{\mathbf{V}}$ denotes the $2D \times 1$ *real-valued* RF vector

$$\mathbf{z}_{\mathbf{V}}(\mathbf{a}) = D^{-\frac{1}{2}} \quad (6)$$
$$\times \left[\sin(\mathbf{v}_1^\top \mathbf{a}), \dots, \sin(\mathbf{v}_D^\top \mathbf{a}), \cos(\mathbf{v}_1^\top \mathbf{a}), \dots, \cos(\mathbf{v}_D^\top \mathbf{a}) \right]^\top.$$

Hence, the nonlinear function that is optimal in the sense of (1) can be approximated by a linear one in the $2D$-dimensional RF space (cf. (4) and (5))

$$\hat{f}^{\mathrm{RF}}(\mathbf{a}) = \sum_{m=1}^{M} \alpha_m \mathbf{z}_{\mathbf{V}}^\top(\mathbf{a}_m) \mathbf{z}_{\mathbf{V}}(\mathbf{a}) := \boldsymbol{\theta}^\top \mathbf{z}_{\mathbf{V}}(\mathbf{a}) \quad (7)$$

where $\boldsymbol{\theta}^\top := \sum_{m=1}^{M} \alpha_m \mathbf{z}_{\mathbf{V}}^\top(\mathbf{a}_m)$. While $\hat{f}$ is the superposition of nonlinear functions κ, its RF approximant $\hat{f}^{\mathrm{RF}}$ in (7) is a linear function of $\mathbf{z}_{\mathbf{V}}(\mathbf{a})$. Note that the dimension of variable $\boldsymbol{\theta}$ is $2D$, which does not depend on M.

Given a network of N nodes, letting v_t denote the node sampled at the tth time slot, and having available $\{\mathbf{a}_t, y_t\}$ at v_t, the inference task over T snapshots can be written as [c.f. (2),(7)]

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{2D}} \sum_{t=1}^{T} \mathcal{L} \left(\boldsymbol{\theta}^\top \mathbf{z}_{\mathbf{V}}(\mathbf{a}_t), y_t \right) \quad (8)$$
$$\mathcal{L}\big(\boldsymbol{\theta}^\top \mathbf{z}_{\mathbf{V}}(\mathbf{a}_t), y_t\big) := \mathcal{C}\big(\boldsymbol{\theta}^\top \mathbf{z}_{\mathbf{V}}(\mathbf{a}_t), y_t\big) + \mu \Omega\big(\|\boldsymbol{\theta}\|^2\big)$$

where $\|\boldsymbol{\theta}\|^2 := \sum_t \sum_\tau \alpha_t \alpha_\tau \mathbf{z}_{\mathbf{V}}^\top(\mathbf{a}_t) \mathbf{z}_{\mathbf{V}}(\mathbf{a}_\tau) := \|f\|_{\mathcal{H}}^2$.

3.2. Online RF-based learning over graphs

In the present section, we will leverage RF-based learning over graphs to enable real-time learning of signals evolving over possibly dynamic networks. A scalable online algorithm will be introduced, which can sequentially sample nodal features and update the sought function estimates.

Training sequentially. In the training phase, we are given a network of N nodes, and the nodal function is sampled in a sequential fashion. Letting v_t denote the node sampled at the tth time slot, and having available $\{\mathbf{a}_t, y_t\}$ at v_t, the RF of its connectivity pattern $\mathbf{z}_{\mathbf{V}}(\mathbf{a}_t)$ is formed as in (6), and $\boldsymbol{\theta}_{t+1}$ is updated 'on the fly,' as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla \mathcal{L}(\boldsymbol{\theta}_t^\top \mathbf{z}_{\mathbf{V}}(\mathbf{a}_t), y_t) \tag{9}$$

where $\{\eta_t\}$ is the sequence of stepsizes that can tune learning rates. In this paper, we will adopt $\eta_t = \eta$ for simplicity. Iteration (9) provides *a functional update* since $\hat{f}_t^{\mathrm{RF}}(\mathbf{a}) = \boldsymbol{\theta}_t^\top \mathbf{z}_{\mathbf{V}}(\mathbf{a})$. Clearly, the per iteration complexity does not increase with the number of nodes N, hence it is scalable with the network size.

Newly-joining nodes. When new nodes join the network, batch graph-kernel based approaches must expand $\bar{\mathbf{K}}$ in (3) by one row and one column, and re-solve (2) in order to form signal estimates for the newly-joining nodes. Hence, each newly joining node will incur complexity $\mathcal{O}(N^3)$. The novel online RF method on the other hand, can simply estimate the signal on the newly coming node via $\hat{f}(v_{\mathrm{new}}) = \hat{\boldsymbol{\theta}} \mathbf{z}_{\mathbf{V}}(\mathbf{a}_{\mathrm{new}})$, where $\mathbf{a}_{\mathrm{new}}$ denotes the connectivity pattern of the new node with the existing nodes in the network. This leads to a complexity of $\mathcal{O}(ND)$ per new node.

Remark 1 (Privacy). The update in (9) does not require access to $\mathbf{a}_t$ directly. Instead, the only information each node needs to reveal is $\mathbf{z}_{\mathbf{V}}(\mathbf{a}_t)$ for each $\mathbf{a}_t$, which involves $\{\sin(\mathbf{a}_t^\top \mathbf{v}_j),\ \cos(\mathbf{a}_t^\top \mathbf{v}_j)\}_{j=1}^D$. These co-sinusoids can be viewed as an encryption of the nodal connectivity pattern.

4. ONLINE GRAPH-ADAPTIVE MKL

In the present section, we develop an online **gr**aph-**ad**aptive learning approach that relies on **ra**ndom features, and leverages multi-**ker**nel approximation to estimate the f based on sequentially obtained nodal samples over the graph. The proposed method is henceforth abbreviated as **Gradraker**.

Note that the choice of κ is critical for the performance of single kernel based learning over graphs, since different kernels capture different properties of the graph, and thus lead to function estimates of variable accuracy. To deal with this, we will assume that the sought function is of the form $f(v) = f(\mathbf{a}) := \sum_{p=1}^P \bar{w}_p f_p(\mathbf{a})$, where $f := \bar{f}/\sum_{p=1}^P w_p$, and the normalized weights $\{\bar{w}_p := w_p/\sum_{p=1}^P w_p\}_{p=1}^P$ satisfy $\bar{w}_p \geq 0$, and $\sum_{p=1}^P \bar{w}_p = 1$.

Given the connectivity pattern $\mathbf{a}_t$ of the tth sampled node v_t, an RF vector $\mathbf{z}_p(\mathbf{a}_t)$ is generated per p from the pdf $\pi_{\kappa_p}(\mathbf{v})$ via (6), where $\mathbf{z}_p(\mathbf{a}_t) := \mathbf{z}_{\mathbf{V}_p}(\mathbf{a}_t)$ for notational brevity. Hence, per kernel κ_p and node sample t, we have $\hat{f}_{p,t}^{\mathrm{RF}}(\mathbf{a}_t) = \boldsymbol{\theta}_{p,t}^\top \mathbf{z}_p(\mathbf{a}_t)$, and as in (9), $\boldsymbol{\theta}_{p,t}$ is updated via

$$\boldsymbol{\theta}_{p,t+1} = \boldsymbol{\theta}_{p,t} - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_{p,t}^\top \mathbf{z}_p(\mathbf{a}_t), y_t) \tag{10}$$

with $\eta \in (0,1)$ chosen constant to effect the adaptation, and $\mathcal{L}_t(\hat{f}_p^{\mathrm{RF}}(\mathbf{a}_t)) := \mathcal{L}\big(\boldsymbol{\theta}^\top \mathbf{z}_{\mathbf{V}_p}(\mathbf{a}_t), y_t\big)$. As far as $\bar{w}_{p,t}$ is concerned, since it resides on the probability simplex, a multiplicative update is well motivated as discussed also in, e.g., [19, 20, 21]. For the un-normalized weights, this update is available in closed form as

$$w_{p,t+1} = w_{p,t} \exp\left(-\eta \mathcal{L}_t\left(\hat{f}_{p,t}^{\mathrm{RF}}(\mathbf{a}_t)\right)\right). \tag{11}$$

Having found $\{w_{p,t}\}$ as in (11), the normalized weights are obtained as $\bar{w}_{p,t} := w_{p,t}/\sum_{p=1}^P w_{p,t}$, and the function estimate can henceforth be obtained by $\hat{f}_{t+1}^{\mathrm{RF}}(\mathbf{a}_{t+1}) = \sum_{p=1}^P \bar{w}_{p,t+1} \hat{f}_{p,t+1}^{\mathrm{RF}}(\mathbf{a}_{t+1})$. Note from (11) that when $\hat{f}_{p,t}^{\mathrm{RF}}$ has a larger loss relative to other $\hat{f}_{p',t}^{\mathrm{RF}}$ with $p' \neq p$ for the tth sampled node, the corresponding $w_{p,t+1}$ decreases more than the other weights. In other words, a more accurate approximant tends to play a more important role in predicting the ensuing sampled node.

4.1. Performance analysis

In order to quantify the performance of Gradraker, we resort to the static regret metric, which quantifies the difference between the aggregate loss of an OCO algorithm, and that of the best fixed function approximant in hindsight, see also e.g., [22, 19]. We establish the regret of our Gradraker approach in the following lemma.

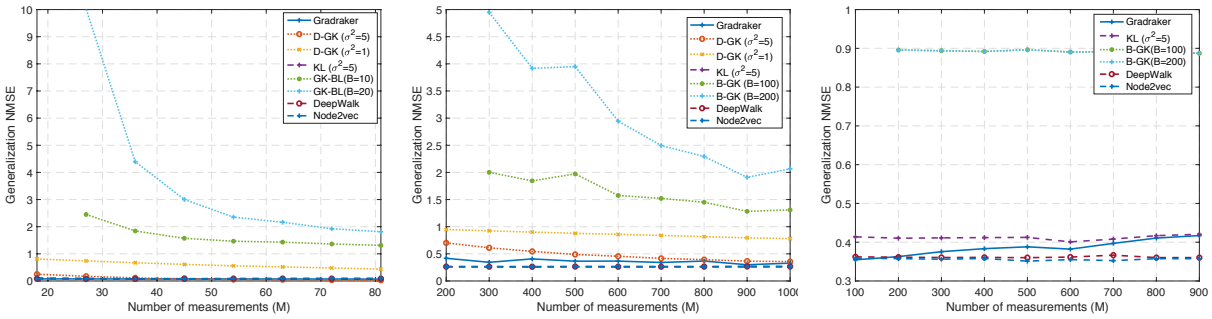
Lemma 1 *With* $\hat{f}_p^*(\cdot) \in \arg\min_{f \in \hat{\mathcal{F}}_p} \sum_{t=1}^T \mathcal{L}_t(f(\mathbf{a}_t))$*, and* $\hat{\mathcal{F}}_p := \{\hat{f}_p | \hat{f}_p(\mathbf{a}) = \boldsymbol{\theta}^\top \mathbf{z}_p(\mathbf{a}),\ \forall \boldsymbol{\theta} \in \mathbb{R}^{2D}\}$*, for any* p*, the sequences* $\{\hat{f}_{p,t}\}$ *and* $\{\bar{w}_{p,t}\}$ *generated by Gradraker satisfy the following bound*

$$\sum_{t=1}^T \mathcal{L}_t\Big(\sum_{p=1}^P \bar{w}_{p,t} \hat{f}_{p,t}(\mathbf{a}_t)\Big) - \sum_{t=1}^T \mathcal{L}_t(\hat{f}_p^*(\mathbf{a}_t))$$
$$\leq \frac{\ln P}{\eta} + \frac{\|\boldsymbol{\theta}_p^*\|^2}{2\eta} + \frac{\eta L^2 T}{2} + \eta T \tag{12}$$

where $\boldsymbol{\theta}_p^*$ *is associated with the best RF function approximant* $\hat{f}_p^*(\mathbf{a}) = \big(\boldsymbol{\theta}_p^*\big)^\top \mathbf{z}_p(\mathbf{a})$.

5. EXPERIMENTS

In this section, Gradraker is tested on three real datasets to corroborate its effectiveness. Due to space limitations, only regression tests are included. In each experiment, the overall network contains N_a nodes, M of which are selected at



(a) (b) (c)

Fig. 1. NMSE performance: a) Temperature; b) Cora; and, c) E-mail.

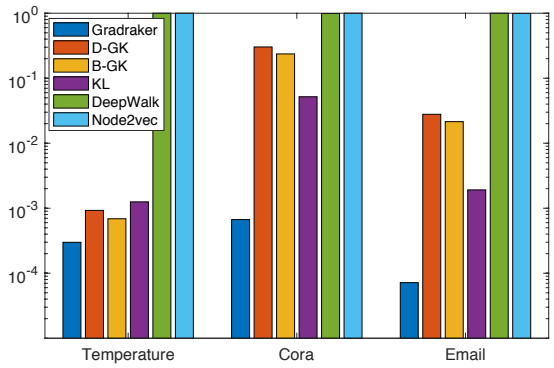


Fig. 2. Normalized runtime comparison.

random, and are treated as a given initial graph with $N = M$ nodes, while the remaining $N_a - N$ nodes are treated as newly-joining nodes.The latter have function values and connectivity unknown at the training phase. The runtime for estimating the function values on the newly-joining nodes, as well as the generalization $\text{NMSE} := \|\hat{\mathbf{y}}_{\mathcal{S}^c} - \mathbf{y}_{\mathcal{S}^c}\|_2^2 / \|\mathbf{y}_{\mathcal{S}^c}\|_2^2$ performance is evaluated, where $\mathcal{S}^c$ denotes the index set of new nodes. The Gradraker is compared with: a) the diffusion graph kernel (D-GK) based method using diffusion kernels with different bandwidths, or band-limited graph kernels (B-GK) with different bandwidths; b) kernel based learning (KL) without RF approximation; c) DeepWalk; and d) Node2vec. Results are averaged over 100 independent runs. The parameters for all algorithms are selected via cross validation. The test results are averaged over 30 independent runs with randomly sampled nodes.

Datasets. *Temperature dataset* comprises 24 time series corresponding to the average temperature per month measured by 89 stations in Switzerland [23]. The graph is constructed based on historical temperature data. Each station is represented by a node, and the temperature at each station is the graph function to be estimated. *Cora dataset* contains $2,708$ scientific publications. Each node here corresponds to a publication, and each of the $5,429$ links connects node i to node j, if paper i cites paper j [5]. The goal is to infer the category each paper belongs to. *Email dataset* consists of $N = 1,005$ nodes, and $25,571$ edges. Each node represents a person, and an edge (i, j) is present if person i sent person j at least one email [24]. The nodal value is the label of the department a person belongs to.

Performance Gradraker adopts a dictionary consisting of 3 Gaussian kernels with parameters $\sigma^2 = 1, 5, 10$. It relies on $D = 100$ random features for the temperature dataset, and $D = 20$ for the E-mail and Cora datasets.

Fig. 1 compares the performance of Gradraker with those of the competing alternatives. It can be readily observed that Gradraker outperforms batch single kernel based approaches with a large margin in terms of NMSE in all three datasets. While Gradraker's NMSE is slightly higher in the E-mail dataset, it is comparable with DeepWalk and NodeVec in the Temperature and Cora datasets. Figure 2 shows the performance of the normalized CPU runtime when new nodes join the network of competitive algorithms. It can be observed that Gradraker is significantly faster than competing alternatives.

6. CONCLUSIONS

The present paper dealt with inference of functions defined over graphs using function values over a subset of nodes. An online MKL-based algorithm was developed, which is capable of tracking nodal functions even when samples are collected sequentially. The novel scheme is highly scalable, and can estimate the unknown function values on newly joining nodes. Moreover, it only relies on encrypted nodal connectivity information. This work opens up a number of interesting directions for future research, including: a) distributed implementations that are well motivated in large-scale networks; b) graph-adaptive learning when multiple sets of nodal features are also available; and c) development of adaptive sampling strategies for Gradraker.

7. REFERENCES

[1] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. Springer, 2009.

[2] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *Proc. Intl. Conf. on Machine Learning*, Sydney, Australia, Jul. 2002, pp. 315–322.

[3] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. of Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.

[4] L. Wasserman and J. D. Lafferty, "Statistical analysis of semi-supervised regression," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2008, pp. 801–808.

[5] Q. Lu and L. Getoor, "Link-based classification," in *Proc. of Intl. Conf. on Machine Learning*, Washington DC, USA, 2003, pp. 496–503.

[6] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proc. of the IEEE*, vol. 106, no. 5, pp. 787–807, May 2018.

[7] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 274–288, Jan. 2017.

[8] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc IEEE Intl. Conf. Acoust. Speech Signal Process.*, Vancouver, Canada, 2013, pp. 5445–5449.

[9] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2432–2444, May 2015.

[10] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. on Sig. Process.*, vol. 65, no. 3, pp. 764–778, Feb. 2017.

[11] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1832–1843, April 2016.

[12] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Sig. Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[13] A. J. Smola and R. I. Kondor, "Kernels and regularization on graphs," in *Learning Theory and Kernel Machines*. Springer, 2003, pp. 144–158.

[14] V. N. Ioannidis, Y. Shen, and G. B. Giannakis, "Semi-blind inference of topologies and dynamical processes over dynamic graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2263–2274, 2019.

[15] G. Wahba, *Spline Models for Observational Data*. Philadelphia, PA: SIAM, 1990.

[16] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Advances in Neural Info. Process. Syst.*, Vancouver, Canada, Dec. 2007, pp. 1177–1184.

[17] Y. Shen, T. Chen, and G. B. Giannakis, "Online ensemble multi-kernel learning adaptive to non-stationary and adversarial environments," in *Proc. of Intl. Conf. on Artificial Intelligence and Statistics*, Lanzarote, Canary Islands, Apr. 2018.

[18] Y. Shen, G. Leus, and G. B. Giannakis, "Online graph-adaptive learning with scalability and privacy," *IEEE Trans. Signal Processing*, vol. 67, no. 9, pp. 2471–2483, May 2019.

[19] E. Hazan, "Introduction to online convex optimization," *Found. and Trends in Mach. Learn.*, vol. 2, no. 3-4, pp. 157–325, 2016.

[20] Y. Shen, T. Chen, and G. B. Giannakis, "Random feature-based online multi-kernel learning in environments with unknown dynamics," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 773–808, 2019.

[21] D. Sahoo, S. C. Hoi, and B. Li, "Online multiple kernel regression," in *Proc. Intl. Conf. Knowledge Discovery and Data Mining*, New York, NY, Aug. 2014, pp. 293–302.

[22] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. and Trends in Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2011.

[23] "Meteorology and climatology meteoswiss." [Online]. Available: http://www.meteoswiss.admin.ch/home/climate/past/climate-normals/climate-diagrams-and-normal-values-per-station.html

[24] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, Mar. 2007.