Online Graph-Guided Inference Using Ensemble Gaussian Processes of Egonet Features

Konstantinos D. Polyzos, Qin Lu, and Georgios B. Giannakis

Department of Electrical and Computer Engineering, University of Minnesota, USA

2021 55th Asilomar Conference on Signals, Systems, and Computers | 978-1-6654-5828-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IEEECONF53345.2021.9723225

***Abstract*—Graph-guided semi-supervised learning (SSL) and inference has emerged as an attractive research field thanks to its documented impact in a gamut of application domains, including transportation and power networks, biological, social, environmental, and financial ones. Distinct from SSL approaches that yield point estimates of the variables to be inferred, the present work puts forth a Bayesian interval learning framework that utilizes Gaussian processes (GPs) to allow for *uncertainty quantification* – a key component in safety-critical applications. An ensemble (E) of GPs is employed to offer an expressive model of the learning function that is updated *incrementally* as nodal observations become available – what caters also for delay-sensitive settings. For the first time in graph-guided SSL and inference, *egonet features* per node are utilized as input to the EGP learning function to account for higher order interactions than the one-hop connectivity of each node. Further enhancing these attributes through random features that encrypt sensitive information per node offers scalability and privacy for the EGP-based learning approach. Numerical tests on real and synthetic datasets corroborate the effectiveness of the novel method.**

***Index Terms*—Gaussian processes, ensemble learning, online learning, egonet features, semi-supervised learning over graphs**

I. Introduction

In the last decade, graph-guided semi-supervised learning (SSL) has gained popularity because of its impact in a number of network science applications, including biological, social, as well as transportation networks to list a few [6]. Scarcity in nodal samples emanating from privacy concerns or sampling costs, gives rise to the SSL task that aims at reconstructing unobserved nodal values based on these limited nodal observations. For instance, in a social network with nodes representing users and edges their relations, a user may be disinclined to reveal private information such as political beliefs, salary, or age.

Several deterministic SSL approaches capitalize on the notion of 'smoothness,' which intuitively suggests that neighboring nodes should have similar nodal values. Such smoothness is manifested via graph kernels [9], [29], [23], [14], [13], [12], Gauss-Markov random fields [34], or, low-rank parametric models [27]. Other existing approaches rely on 'graph bandlimitedness' [24], sparsity, or, overcomplete dictionaries [5]. In recent years, graph neural network-based approaches, which generalize convolution operators to graph-structured data, have also gained popularity because of their appealing performance in certain domains; see, e.g., [7], [11], [33]. Albeit interesting, the aforementioned approaches require storing the connectivity patterns of all nodes. To deal with this limitation, a recent multi-kernel online approach reconstructs nodal values relying on their one-hop connectivity vectors [26]. The per-node one-hop connectivity vector reveals relational information between the node itself and its direct neighbors, which may have limited representational power, and can compromise prediction performance. In addition, although scalable in computational and storage demand, *deterministic* SSL approaches, including [26], do not quantify the associated uncertainty, which is instrumental especially in settings, where safety is a prime concern.

Gaussian processes (GPs) offer a nonparametric Bayesian model, where the sought learning function is viewed as random, and can be fully characterized by its probability density function (pdf) [22]. In the graph-guided SSL context, GPs have been adopted through graph kernels that hinge on the entire adjacency matrix; see, e.g. [17], [4], [28], [32], [31]. These typically *batch* approaches though require storing relational information of the entire graph, which may become prohibitive for large networks. Most recently, an online and scalable GP development has been devised in [19] that relies on the per-node one-hop connectivity vector, but does not account for interactions beyond the single-hop ones.

Contributions. In this work, we put forth a Bayesian graph-guided SSL approach that relies on GPs and offers extra uncertainty quantification, which is of paramount importance in safety-critical applications. Accounting for higher order interactions compared to the per-node one-hop connectivity vector adopted in [19], [26], the proposed framework leverages the so-termed egonet features to form the input vector of the learning function. Although the notion of "egonet" has been utilized in several graph-related tasks, such as anomaly detection [2] and community detection [25], this is the first time egonets are employed in the SSL context. Capitalizing on random features to effect *scalability* and preserve *privacy*, an ensemble (E) of graph-adaptive GP experts is adopted to learn a more expressive function incrementally, where nodal values are predicted and processed in a streaming fashion to significantly save data storage, especially for larger networks. Meanwhile, the proposed framework adapts to the appropriate EGP kernels on-the-fly bypassing the need for a pre-selected kernel and offline training as in conventional GPs. Experiments conducted on real and synthetic graph datasets demonstrate the effectiveness of the advocated EGP approach.

This work was supported in part by ARO grant W911NF2110297, and NSF grants 2126052 and 1901134. The work of Konstantinos D. Polyzos was also supported by the Onassis Foundation Scholarship.

Emails: {*polyz003, qlu, georgios*}*@umn.edu*

II. Preliminaries and Problem Formulation

Consider a graph $\mathcal{G} := \{\mathcal{V}, \mathbf{A}\}$, where the vertex set $\mathcal{V} := \{1, \ldots, N\}$ collects all N nodes, and the $N \times N$ adjacency matrix $\mathbf{A}$ captures the nodal connectivity patterns. The (n, n')th entry of $\mathbf{A}$, namely $a_{nn'} := \mathbf{A}(n, n')$, represents the weighted link connecting nodes n and n'. A real-valued function or signal on the graph is given by the mapping $f : \mathcal{V} \to \mathbb{R}$ with f_n denoting the true feature value at node n, which is then mapped to the observed nodal value y_n. For example, f_n could represent the annual income or political alignment of user n in a social network.

Given $\mathcal{G}$ and a subset of nodal observations $\{y_n, n \in \mathcal{O}\}$, where $\mathcal{O}$ collects the indices of measured nodes, SSL aims at predicting nodal values (or labels) of unobserved nodes $\{y_n, n \in \mathcal{U}\}$, where $\mathcal{U} := \mathcal{V} \setminus \mathcal{O}$. This extrapolation task can be performed either in a typical *batch* setting, or, in an online *incremental* mode, where past observations $\mathbf{y}_n := [y_1, \ldots, y_n]^\top$ are used to form the predictor $\hat{y}_{n+1}$, before the new datum y_{n+1} becomes available, and subsequently y_{n+1} is processed to aid future predictions. Unlike batch approaches, the incremental scheme significantly saves data storage especially for large networks, and is well motivated in delay-sensitive applications, as well as in cases where the data are *non-stationary* or *adversarially* chosen. This incremental setting has been considered in [26] and [19], where f_n is mapped from node n's one-hop connectivity vector $\mathbf{a}_n := \mathbf{A}(:, n)$ that can have limited expressiveness. Accounting for higher-order interactions relative to $\mathbf{a}_n$, the present work advocates the novel adoption of the so-termed "egonet" of each node in order to form the input vector.

A. Nodal egonet features

The egonet of node n is defined as the subgraph that consists of node n itself, its direct neighbors, and all the edges connecting them from the original edge set. Let $\mathbf{A}_n^{\text{ego}}$ denote the adjacency matrix that describes the egonet of node n, where all entries concerning nodes that do not belong to the egonet are set to 0. An illustrative example of a node's egonet is provided in Fig. 1. Relying on $\mathbf{A}_n^{\text{ego}}$, the per-node "egonet feature" vector $\mathbf{x}_n^{\text{ego}}$ will be constructed to summarize characteristics about node n's egonet. In the present work, $\mathbf{x}_n^{\text{ego}}$ is chosen to capture the importance of all existing nodes in the egonet through the notion of "vertex centrality" [8]. The simplest form of vertex centrality is the degree d_n of node n, which is defined as $d_n := \sum_{n'=1}^{N} \mathbf{A}_n^{\text{ego}}(n', n)$. Further measuring the significance of a node based on the impact of its neighbors yields the well-known eigenvector centrality, whose value for node m is given by

$$c_{\text{Ei}}^n(m) = \alpha \sum_{m' \in \mathcal{N}_m^n} c_{\text{Ei}}^n(m') \tag{1}$$

where $\mathcal{N}_m^n$ contains all existing neighbors of node m in egonet $\mathbf{A}_n^{\text{ego}}$. If the values of eigenvector centrality across all N nodes are collected in $\mathbf{c}_{\text{Ei}}^n := [c_{\text{Ei}}^n(1), \ldots, c_{\text{Ei}}^n(N)]^\top$, it can be readily seen that $\mathbf{c}_{\text{Ei}}^n$ and α can be obtained by solving the eigendecomposition problem[1] $\mathbf{A}_n^{\text{ego}} \mathbf{c}_{Ei} = \alpha^{-1} \mathbf{c}_{Ei}$, where α^{-1} is selected to be the largest eigenvalue of $\mathbf{A}_n^{\text{ego}}$ and $\mathbf{c}_{\text{Ei}}^n$ is the associated eigenvector [3]. Note that besides the degree and the eigenvector centrality, the egonet feature vector $\mathbf{x}_n^{\text{ego}}$ can comprise additional characteristics of the egonet $\mathbf{A}_n^{\text{ego}}$, including edge centrality, the clustering coefficient or the network cohesion, to list a few; refer to [8, Chapter 5] for an overview of these metrics.

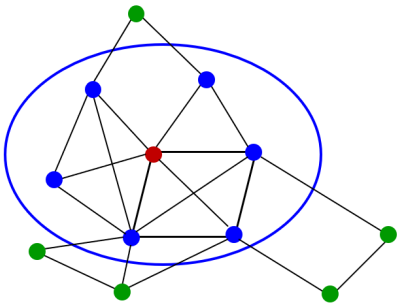

Fig. 1: Egonet of the red node consists of itself, its one-hop neighbors (blue nodes), and all edges connecting them.

B. GP-based learning over graphs

With the egonet feature vector $\mathbf{x}_n^{\text{ego}}$ at hand, the aforementioned extrapolation task will be addressed via Gaussian processes (GPs) that are attractive because they offer non-parametric function estimates with quantifiable uncertainty [22]. Different from deterministic approaches, such as [26], the unknown f is modeled as random with a GP prior as $f \sim \mathcal{GP}(0, \kappa(\mathbf{x}^{\text{ego}}, \mathbf{x}'^{\text{ego}}))$, where $\kappa(\cdot, \cdot)$ is a positive-definite kernel function that measures pairwise similarity between $\mathbf{x}^{\text{ego}}$ and $\mathbf{x}'^{\text{ego}}$. This means that for any number of inputs $\mathbf{X}_n := [\mathbf{x}_1^{\text{ego}}, \ldots, \mathbf{x}_n^{\text{ego}}]$, the prior pdf of the $n \times 1$ function evaluations $\mathbf{f}_n := [f(\mathbf{x}_1^{\text{ego}}), \ldots, f(\mathbf{x}_n^{\text{ego}})]^\top$ is jointly Gaussian

$$p(\mathbf{f}_n; \mathbf{X}_n) = \mathcal{N}(\mathbf{f}_n; \mathbf{0}_n, \mathbf{K}_n) \quad \forall n \tag{2}$$

where $\mathbf{K}_n$ is the $n \times n$ covariance matrix whose (i, j)th entry is $[\mathbf{K}_n]_{ij} = \text{cov}(f(\mathbf{x}_i^{\text{ego}}), f(\mathbf{x}_j^{\text{ego}})) := \kappa(\mathbf{x}_i^{\text{ego}}, \mathbf{x}_j^{\text{ego}})$.

Nodal observations $\mathbf{y}_n := [y_1, \ldots, y_n]^\top$ are related to the function evaluations $\mathbf{f}_n$ via the conditional likelihood $p(\mathbf{y}_n | \mathbf{f}_n; \mathbf{X}_n) = \prod_{n'=1}^{n} p(y_{n'} | f(\mathbf{x}_{n'}^{\text{ego}}))$, that is presumed factorable over the per-datum likelihoods $p(y_{n'} | f(\mathbf{x}_{n'}^{\text{ego}})) = \mathcal{N}(y_{n'}; f(\mathbf{x}_{n'}^{\text{ego}}), \sigma_\varepsilon^2)$ with known variance σ_ε^2. This factorization certainly holds when $y_n = f_n + \varepsilon_n$, and noise $\varepsilon_n \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is uncorrelated across nodes.

The GP prior and Gaussian likelihood yield the following Gaussian pdf for $\mathbf{y}_n$ and observation y_{n+1} of a new node with egonet features $\mathbf{x}_{n+1}^{\text{ego}}$ [22]

$$\begin{bmatrix} \mathbf{y}_n \\ y_{n+1} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}_{n+1}, \begin{bmatrix} \mathbf{K}_n + \sigma_\varepsilon^2 \mathbf{I}_n & \mathbf{k}_{n+1} \\ \mathbf{k}_{n+1}^\top & \kappa(\mathbf{x}_{n+1}^{\text{ego}}, \mathbf{x}_{n+1}^{\text{ego}}) + \sigma_\varepsilon^2 \end{bmatrix} \right)$$

[1] Targeting at a positive definite matrix, $\mathbf{A}_n^{\text{ego}} \mathbf{A}_n^{\text{ego}\top}$ can alternatively be used to form the eigenvector centrality based egonet feature vector.

where $\mathbf{k}_{n+1} := [\kappa(\mathbf{x}_1^{\text{ego}}, \mathbf{x}_{n+1}^{\text{ego}}), \ldots, \kappa(\mathbf{x}_n^{\text{ego}}, \mathbf{x}_{n+1}^{\text{ego}})]^\top$, and $\mathbf{I}_n$ is the $n \times n$ identity matrix. Hence, the predictive pdf of y_{n+1} conditioned on $\mathbf{y}_n$ is

$$p(y_{n+1}|\mathbf{y}_n; \mathbf{X}_{n+1}) = \mathcal{N}(y_{n+1}; \hat{y}_{n+1|\mathbf{n}}, \sigma^2_{n+1|\mathbf{n}})$$

where bold $_{\mathbf{n}}$ marks that all n data are used to form

$$\hat{y}_{n+1|\mathbf{n}} = \mathbf{k}_{n+1}^\top (\mathbf{K}_n + \sigma_\varepsilon^2 \mathbf{I}_n)^{-1} \mathbf{y}_n \tag{3a}$$

$$\sigma^2_{n+1|\mathbf{n}} = \kappa(\mathbf{x}_{n+1}^{\text{ego}}, \mathbf{x}_{n+1}^{\text{ego}}) - \mathbf{k}_{n+1}^\top (\mathbf{K}_n + \sigma_\varepsilon^2 \mathbf{I}_n)^{-1} \mathbf{k}_{n+1} + \sigma_\varepsilon^2 \,. \tag{3b}$$

Note that the mean (3a) provides a point prediction of y_{n+1} and the variance (3b) quantifies the associated uncertainty of this prediction.

Albeit interesting, this extrapolation approach requires inversion of an $n \times n$ kernel matrix, which incurs complexity $\mathcal{O}(n^3)$, rendering it non-scalable for large values of n. In addition, the performance of the GP-based predictor in (3a) relies on the preselected kernel κ, which calls for offline model training. In the rest of the paper, a scalable approach will be developed and assessed. Its upshot is that it can adapt to the appropriate kernel on-the-fly as graph data arrive in a streaming fashion, thus bypassing the need for both offline kernel pre-selection and data storage.

III. Incremental graph-guided EGP learning

Aiming to enrich the function space and adapt kernel(s) on-the-fly, we employ an ensemble of M GP experts [16], [18], [15], each relying on a unique kernel, chosen from a known dictionary $\mathcal{K} := \{\kappa^1, \ldots, \kappa^M\}$. With per-node egonet features as input, each expert m postulates a GP prior on f, denoted as $f|m \sim \mathcal{GP}(0, \kappa^m(\mathbf{x}^{\text{ego}}, \mathbf{x}'^{\text{ego}}))$, which means that the per-expert prior pdf of $\mathbf{f}_n$ is $p(\mathbf{f}_n|m; \mathbf{X}_n) = \mathcal{N}(\mathbf{f}_n; \mathbf{0}_n, \mathbf{K}_n^m)$ (cf. (2)). Considering all M experts, the ensemble (E) GP prior gives rise to the following Gaussian mixture (GM) pdf

$$p(\mathbf{f}_n; \mathbf{X}_n) = \sum_{m=1}^M w^m \mathcal{N}(\mathbf{f}_n; \mathbf{0}_n, \mathbf{K}_n^m) \,, \quad \sum_{m=1}^M w^m = 1 \tag{4}$$

where the weights $\{w^m\}_{m=1}^M$ ($w^m \in [0,1]$) can be viewed as probabilities capturing the significance of the GP experts in the EGP meta-learner. Considering an incremental setting, where nodal observations arrive in a streaming fashion, these weights are adapted as $w_n^m := \Pr(m|\mathbf{y}_n; \mathbf{X}_n)$.

In the remainder of this section, we introduce our EGP approach to incremental graph-guided learning, where each GP expert is endowed with scalability using an approximate parametric so-termed random feature model.

A. Random feature approximation for scalability

Existing scalable GP-based approaches rely on structured approximants of the kernel matrix; see, e.g., [30], [10]. In this work, each expert m leverages a shift-invariant kernel $\kappa^m(\mathbf{x}, \mathbf{x}')$ to construct a low-rank approximant of $\mathbf{K}_n^m$. If $\bar{\kappa}^m := \kappa^m / \sigma^2_{\theta^m}$ denotes the standardized version of κ^m with $\sigma^2_{\theta^m}$ such that $\bar{\kappa}^m(0) = 1$, then the inverse Fourier transform of $\bar{\kappa}^m$ yields

$$\bar{\kappa}^m(\mathbf{x}, \mathbf{x}') = \bar{\kappa}^m(\mathbf{x} - \mathbf{x}') = \int \pi^m_{\bar{\kappa}}(\mathbf{v}) e^{j\mathbf{v}^\top(\mathbf{x}-\mathbf{x}')} d\mathbf{v}$$
$$:= \mathbb{E}_{\pi^m_{\bar{\kappa}}} \left[e^{j\mathbf{v}^\top(\mathbf{x}-\mathbf{x}')} \right] \tag{5}$$

where $\pi^m_{\bar{\kappa}}$ is the power spectral density, which is normalized in order to integrate to 1, so that it can be viewed as a pdf. Upon drawing D vectors $\{\mathbf{v}_i^m\}_{i=1}^D$ independently from $\pi^m_{\bar{\kappa}}$, $\bar{\kappa}^m(\mathbf{x}, \mathbf{x}')$ can be approximated by the sample average $\check{\bar{\kappa}}^m(\mathbf{x}, \mathbf{x}') = D^{-1} \sum_{i=1}^D \cos(\mathbf{v}_i^{m\top}(\mathbf{x} - \mathbf{x}'))$, where the imaginary part of (5) vanishes because $\bar{\kappa}^m$ is real [21].

Let us now define the $2D \times 1$ random feature (RF) vector

$$\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}) := \tag{6}$$
$$\frac{1}{\sqrt{D}} \left[\sin(\mathbf{v}_1^{m\top}\mathbf{x}), \cos(\mathbf{v}_1^{m\top}\mathbf{x}), \ldots, \sin(\mathbf{v}_D^{m\top}\mathbf{x}), \cos(\mathbf{v}_D^{m\top}\mathbf{x}) \right]^\top$$

based on which $\check{\bar{\kappa}}^m$ can be re-written as $\check{\bar{\kappa}}^m(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}) \boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}')$, and hence each expert m will rely on the random linear approximant $\check{f}$ that obeys the generative *parametric* model

$$p(\check{f}(\mathbf{x}^{\text{ego}})|\boldsymbol{\theta}^m, m) = \delta(\check{f}(\mathbf{x}^{\text{ego}}) - \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}^{\text{ego}}) \boldsymbol{\theta}^m) \tag{7a}$$

$$\boldsymbol{\theta}^m \sim \mathcal{N}(\boldsymbol{\theta}; \mathbf{0}_{2D}, \sigma^2_{\theta^m} \mathbf{I}_{2D}) \,. \tag{7b}$$

It can be readily seen that the prior pdf of $\check{\mathbf{f}}$ is now $p(\check{\mathbf{f}}_n|m; \mathbf{X}_n) = \mathcal{N}(\check{\mathbf{f}}_n; \mathbf{0}_n, \check{\mathbf{K}}_n^m)$, where $\check{\mathbf{K}}_n^m = \sigma^2_{\theta^m} \boldsymbol{\Phi}_n^m \boldsymbol{\Phi}_n^{m\top}$ (with $\boldsymbol{\Phi}_n^m := \left[\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_1^{\text{ego}}), \ldots, \boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_n^{\text{ego}})\right]^\top$) has rank $2D$ when $n > 2D$, which renders it a low-rank approximant of $\mathbf{K}_n^m$.

Based on the parametric form of $\check{f}$ in (7a), the per-expert likelihood is also parameterized by $\boldsymbol{\theta}^m$ as

$$p(y_n|\boldsymbol{\theta}^m, m; \mathbf{x}_n^{\text{ego}}) = \mathcal{N}(y_n; \boldsymbol{\phi}_{\mathbf{v}}^\top(\mathbf{x}_n^{\text{ego}}) \boldsymbol{\theta}^m, \sigma_\varepsilon^2) \tag{8}$$

which, along with the Gaussian prior (7b), yield the posterior

$$p(\boldsymbol{\theta}^m|\mathbf{y}_n, m; \mathbf{X}_n) = \mathcal{N}(\boldsymbol{\theta}^m; \hat{\boldsymbol{\theta}}_n^m, \boldsymbol{\Sigma}_n^m) \tag{9}$$

that is subsequently utilized by expert m to form her/his own prediction. Next, we will show how incremental learning updates $\{\hat{\boldsymbol{\theta}}_n^m, \boldsymbol{\Sigma}_n^m\}_m$, and $\{w_n^m\}_m$ in a data-adaptive fashion, as each new node is considered by our SSL setup.

B. Incremental learning with prediction and correction

Different from [19], the novel graph-guided EGP approach here relies on the per-node egonet features instead of the one-hop connectivity pattern. Abbreviated hereafter as "GradEGP-ego," it proceeds in two steps per node, namely prediction and correction.

Prediction. Before taking into account the nodal value of node $n+1$, expert m constructs the RF vector $\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{n+1}^{\text{ego}})$ via (6), and predicts the pdf of y_{n+1} via the known posterior (9) as

$$p(y_{n+1}|\mathbf{y}_n, m; \mathbf{X}_{n+1})$$
$$= \int p(y_{n+1}|\boldsymbol{\theta}^m, m; \mathbf{x}_{n+1}^{\text{ego}}) p(\boldsymbol{\theta}^m|\mathbf{y}_n, m; \mathbf{X}_n) d\boldsymbol{\theta}^m$$
$$= \mathcal{N}(y_{n+1}; \hat{y}_{n+1|n}^m, (\sigma_{n+1|n}^m)^2) \tag{10}$$

where the predicted mean and variance are

$$\hat{y}_{n+1|n}^m = \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_{n+1}^{\text{ego}}) \hat{\boldsymbol{\theta}}_n^m \tag{11a}$$

$$(\sigma_{n+1|n}^m)^2 = \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_{n+1}^{\text{ego}}) \boldsymbol{\Sigma}_n^m \boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{n+1}^{\text{ego}}) + \sigma_\varepsilon^2 \,. \tag{11b}$$

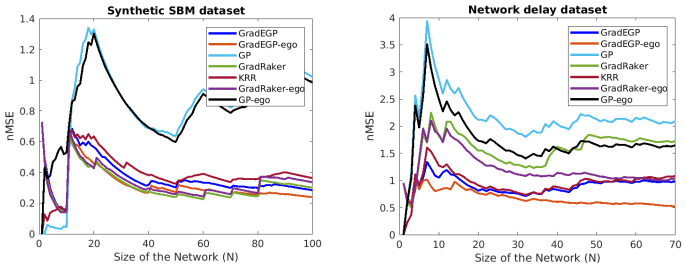

Fig. 2: nMSE performance on (a) "Synthetic SBM;" and (b) "Network delay" datasets.

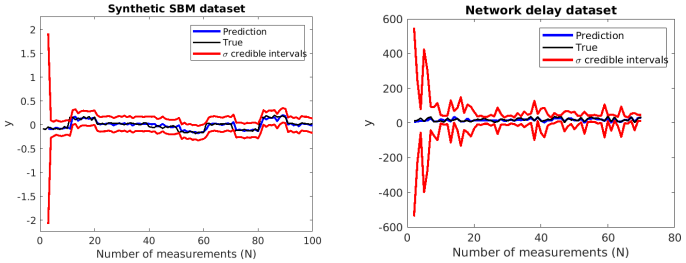

Fig. 3: Uncertainty quantification performance of GradEGP on (a) "Synthetic SBM;" and (b) "Network delay" datasets. Blue lines denote the prediction, black lines the true values, and red lines $\pm\sigma$ confidence intervals.

Accounting for all M experts, the EGP meta-learner predicts the ensemble pdf of y_{n+1}, which is given by the GM

$$p(y_{n+1}|\mathbf{y}_n;\mathbf{X}_{n+1}) = \sum_{m=1}^{M} p(y_{n+1}|\mathbf{y}_n,m;\mathbf{X}_{n+1})p(m|\mathbf{y}_n;\mathbf{X}_n)$$
$$= \sum_{m=1}^{M} w_n^m \mathcal{N}(y_{n+1};\hat{y}_{n+1|n}^m,(\sigma_{n+1|n}^m)^2)\ . \tag{12}$$

Thus, the minimum mean-square error (MMSE) predictor of y_{n+1} along with its corresponding variance are given by

$$\hat{y}_{n+1|n} = \sum_{m=1}^{M} w_n^m \hat{y}_{n+1|n}^m \tag{13a}$$

$$\sigma_{n+1|n}^2 = \sum_{m=1}^{M} w_n^m [(\sigma_{n+1|n}^m)^2 + (\hat{y}_{n+1|n} - \hat{y}_{n+1|n}^m)^2]\ . \tag{13b}$$

Note that the subscript "$n+1|n$" differs from "$n+1|\mathbf{n}$" in the left-hand side of (3), in the sense that the batch GP predictor requires storing the past observations contained in $\mathbf{y}_n$.

Correction. With the arrival of y_{n+1}, expert m updates the weight w_n^m via Bayes' rule as

$$w_{n+1}^m = \Pr(m|\mathbf{y}_{n+1};\mathbf{X}_{n+1}) = \frac{w_n^m p(y_{n+1}|\mathbf{y}_n,m;\mathbf{X}_{n+1})}{p(y_{n+1}|\mathbf{y}_n;\mathbf{X}_{n+1})}$$
$$= \frac{w_n^m \mathcal{N}(y_{n+1};\hat{y}_{n+1|n}^m,(\sigma_{n+1|n}^m)^2)}{\sum_{m'=1}^{M} w_n^{m'} \mathcal{N}(y_{n+1};\hat{y}_{n+1|n}^{m'},(\sigma_{n+1|n}^{m'})^2)}\ . \tag{14}$$

Also, the posterior of $\boldsymbol{\theta}^m$ is propagated via Bayes' rule

$$p(\boldsymbol{\theta}^m|\mathbf{y}_{n+1},m;\mathbf{X}_{n+1}) = \frac{p(\boldsymbol{\theta}^m|\mathbf{y}_n,m;\mathbf{X}_n)p(y_{n+1}|\boldsymbol{\theta}^m,m;\mathbf{x}_{n+1}^{\text{ego}})}{p(y_{n+1}|\mathbf{y}_n,m;\mathbf{X}_{n+1})}$$
$$= \mathcal{N}(\boldsymbol{\theta}^m;\hat{\boldsymbol{\theta}}_{n+1}^m,\boldsymbol{\Sigma}_{n+1}^m) \tag{15}$$

where

$$\hat{\boldsymbol{\theta}}_{n+1}^m = \hat{\boldsymbol{\theta}}_n^m + (\sigma_{n+1|n}^m)^{-2}\boldsymbol{\Sigma}_n^m \boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{n+1}^{\text{ego}})(y_{n+1} - \hat{y}_{n+1|n}^m)$$
$$\boldsymbol{\Sigma}_{n+1}^m = \boldsymbol{\Sigma}_n^m - (\sigma_{n+1|n}^m)^{-2}\boldsymbol{\Sigma}_n^m \boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{x}_{n+1}^{\text{ego}})\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x}_{n+1}^{\text{ego}})\boldsymbol{\Sigma}_n^m\ .$$

With per-node egonet features available, the proposed GradEGP-ego incurs per-iteration complexity $\mathcal{O}(M((2D)^2 + 2DN))$, which means that besides offering a richer function space, GradEGP-ego is also scalable.

Remark (Privacy consideration). The per-expert prediction and correction steps do not entail direct access to the per-node egonet feature vector $\mathbf{x}_n^{\text{ego}}$, but instead capitalize on the RF vector which can be viewed as an encryption of the original input vector. Hence, similar to [26], the advocated GradEGP-ego preserves privacy of nodal information in the graph.

IV. Numerical tests

In order to assess the merits of the novel GradEGP-ego, we conducted tests using both synthetic and real datasets, whose parameters are given next.

Synthetic dataset. A synthetic graph with $N = 100$ nodes is constructed based on the stochastic block model consisting of $C = 10$ communities, as in e.g., [20]. The targeted nodal values are given by the eigenvector corresponding to the lowest nonzero eigenvalue of the graph Laplacian matrix.

Network delay dataset. The measured delays of $N = 70$ paths are considered, each connecting two of 9 end-nodes on the Internet2backbone [1]. A symmetric graph is constructed based on the common links between any two paths, and the sought nodal values are delays experienced on these paths.

The proposed GradEGP-ego approach is compared with its precursor GradEGP in [19] that capitalizes on the per-node one-hop connectivity vector, as well as with three kernel-based approaches, namely GradRaker [26], kernel ridge regression (KRR) [23], and the conventional GP (cf. (3)). The per-node

egonet feature vector is formed by the node's degree and the eigenvector centrality vector $\mathbf{c}_{\text{Ei}}^n$ obtained from $\mathbf{A}_n^{\text{ego}\top}\mathbf{A}_n^{\text{ego}}$, as dicussed in Sec. II-A. Furthermore, we combined our per-node egonet features with Gradraker and vanilla GP, to obtain what we abbreviate as "Gradraker-ego" and "GP-ego," respectively. For all RF-based approaches, we select $D = 50$, and the kernel dictionary consisting of 11 radial basis functions with characteristic length scales $\{10^k\}_{k=-4}^{6}$. For the vanilla GP, we used the best-performing kernel from the EGP prior.

As figure of merit, we used the normalized mean-square error $\text{nMSE}_n := n^{-1}\sum_{n'=1}^{n}(y_{n'} - \hat{y}_{n'|n'-1})^2/s_y^2$, where s_y^2 denotes the sample variance of $\mathbf{y}_N$. As evidenced by Fig. 2, the novel GradEGP-ego consistently outperforms all other alternatives in terms of nMSE, which demonstrates the impact of leveraging higher-order interactions via egonet features. It is worth mentioning that the use of egonet features can also boost the prediction performance of plain-vanilla GP and Gradraker as corroborated by Fig. 2a). Further, the superior performance of EGP based approaches, namely GradEGP-ego and GradEGP, over the batch conventional single-expert GP highlights the benefits of ensembles that can efficiently combine the GP experts by properly adjusting their weights for each prediction. Besides prediction of the nodal values, the novel GradEGP-ego framework additionally offers uncertainty quantification through σ confidence intervals as depicted in Fig. 3, where it is intuitive that the prediction uncertainty deteriorates as the number of observations increases.

V. Conclusions

An ensemble of Gaussian process models was considered in this contribution for graph-guided semi-supervised learning with quantifiable uncertainty. For the first time, the per-node egonet features were adopted as input to the learning function in order to markedly boost the prediction performance by accounting for higher-order node interactions. Leveraging random features for scalability and privacy, an incremental setting was developed to predict the sought nodal values in a streaming fashion, significantly saving data storage. Numerical tests showcased the merits of the proposed method.

References

[1] "One-way ping internet2," http://software.internet2.edu/owamp/, [Online; accessed 29-April-2019].

[2] B. Baingana, P. A. Traganitis, G. B. Giannakis, G. Mateos, and I. Pittas, *Big Data Analytics for Social Networks*. CRC Press, 2016.

[3] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *J. of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[4] W. Chu, V. Sindhwani, Z. Ghahramani, and S. S. Keerthi, "Relational learning with Gaussian processes," *Proc. Advances Neural Inf. Process. Syst.*, pp. 289–296, 2007.

[5] P. A. Forero, K. Rajawat, and G. B. Giannakis, "Prediction of partially observed dynamical processes over networks via dictionary learning," *IEEE Trans. Sig. Process.*, vol. 62, no. 13, pp. 3305–3320, Jul. 2014.

[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] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *Proc. Int. Conf. Learn. Represent.*, Apr. 2017.

[8] E. D. Kolaczyk, *Statistical Analysis of Network Data*. Springer-Verlag New York, 2009.

[9] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," *Proc. Int. Conf. Mach. Learn.*, pp. 315–322, Jul. 2002.

[10] M. Lázaro-Gredilla, J. Quiñonero Candela, C. E. Rasmussen, and A. Figueiras-Vidal, "Sparse spectrum Gaussian process regression," *J. Mach. Learn. Res.*, vol. 11, no. Jun, pp. 1865–1881, 2010.

[11] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. of AAAI Conf. on Artificial Intel.*, New Orleans, Louisiana, Feb. 2018.

[12] Q. Lu and G. B. Giannakis, "Probabilistic reconstruction of spatio-temporal processes over multi-relational graphs," *IEEE Trans. Sig. and Info. Process. over Net.*, vol. 7, pp. 166–176, 2021.

[13] Q. Lu, V. N. Ioannidis, and G. B. Giannakis, "Graph-adaptive semi-supervised tracking of dynamic processes over switching network modes," *IEEE Trans. Sig. Process.*, vol. 68, pp. 2586–2597, 2020.

[14] ——, "Semi-supervised learning of processes over multi-relational graphs," in *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Process.*, 2020, pp. 5560–5564.

[15] Q. Lu, G. Karanikolas, and G. B. Giannakis, "Incremental ensemble Gaussian processes," *arXiv preprint arXiv:2110.06777*, 2021.

[16] Q. Lu, G. Karanikolas, Y. Shen, and G. B. Giannakis, "Ensemble Gaussian processes with spectral features for online interactive learning with scalability," *Proc. Int. Conf. Artificial Intel. and Stats.*, June 2020.

[17] Y. C. Ng, N. Colombo, and R. Silva, "Bayesian semi-supervised learning with graph Gaussian processes," *Proc. Advances Neural Inf. Process. Syst.*, Dec. 2018.

[18] K. D. Polyzos, Q. Lu, and G. B. Giannakis, "Ensemble Gaussian processes for online learning over graphs with adaptivity and scalability," *IEEE Trans. Sig. Process.*, 2021. [Online]. Available: https://doi.org/10.1109/tsp.2021.3122095

[19] ——, "Graph-adaptive incremental learning using an ensemble of Gaussian process experts," *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Process.*, June 2021.

[20] K. D. Polyzos, C. Mavromatis, V. N. Ioannidis, and G. B. Giannakis, "Unveiling anomalous edges and nominal connectivity of attributed networks," *Proc. Asilomar Conf. Sig., Syst., Comput.*, Nov. 2020.

[21] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Proc. Advances Neural Inf. Process. Syst.*, pp. 1177–1184, 2008.

[22] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

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

[24] S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, "Reconstruction of graph signals through percolation from seeding nodes," *IEEE Trans. Sig. Process.*, vol. 64, no. 16, pp. 4363–4378, Aug. 2016.

[25] F. Sheikholeslami and G. B. Giannakis, "Identification of overlapping communities via constrained egonet tensor decomposition," *IEEE Trans. Sig. Process.*, vol. 66, no. 21, pp. 5730–5745, 2018.

[26] Y. Shen, G. Leus, and G. B. Giannakis, "Online graph-adaptive learning with scalability and privacy," *IEEE Trans. Sig. Process.*, vol. 67, no. 9, May 2019.

[27] 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, May 2013.

[28] R. Silva, W. Chu, and Z. Ghahramani, "Hidden common cause relations in relational learning," *Proc. Advances Neural Inf. Process. Syst.*, pp. 1345–1352, 2008.

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

[30] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," *Proc. Int. Conf. Artificial Intel. and Stats.*, pp. 567–574, 2009.

[31] M. Van der Wilk, C. E. Rasmussen, and J. Hensman, "Convolutional Gaussian processes," *Proc. Advances Neural Inf. Process. Syst.*, pp. 2849–2858, 2017.

[32] I. Walker and B. Glocker, "Graph convolutional Gaussian processes," *Proc. Int. Conf. Mach. Learn.*, 2019.

[33] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. on Neural Net. and Learn. Syst.*, 2020.

[34] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," *Proc. Int. Conf. Mach. Learn.*, pp. 912–919, 2003.