Kernel-Based Models for Influence Maximization on Graphs based on Gaussian Process Variance Minimization

Salvatore Cuomo[a], Wolfgang Erb[b,*], Gabriele Santin[c]

[a] *Università degli Studi di Napoli Federico II, Dipartimento di Matematica e Applicazioni "Renato Cacciopoli", Napoli, Italy*
[b] *Università degli Studi di Padova, Dipartimento di Matematica "Tullio Levi-Civita", Padova, Italy*
[c] *Fondazione Bruno Kessler, Digital Society Center, Trento, Italy*

Abstract

The inference of novel knowledge, the discovery of hidden patterns, and the uncovering of insights from large amounts of data from a multitude of sources make data science to an art rather than just a mere scientific discipline. The study and design of mathematical models and signal processing tools able to analyze information represents a central research topic within data science. In this work, we introduce and investigate a model for influence maximization (IM) on graphs using ideas from kernel-based signal approximation, Gaussian process regression, and the minimization of a corresponding variance term. Data-driven approaches can be applied to determine proper kernels for this IM model and machine learning methodologies are adopted to tune the model parameters. Compared to stochastic models for influence maximization that rely on Monte-Carlo simulations, our kernel-based model allows for a simple and cost-efficient update strategy to compute optimal influencing nodes on a graph. In several numerical experiments, we show the properties and benefits of this model.

Keywords: Gaussian process regression on Graphs, Graph Basis Functions (GBFs), Influence Maximization (IM) on Graphs, Kernel-based Inference, Optimal Design of Sampling Nodes, P-greedy Methods, Machine Learning

1. Introduction

Online social platforms such as Facebook, Twitter, LinkedIn, and Tumblr, are pervasively present in our daily life. In these virtual communities, people are emotionally connected and share several kinds of relationships as friendship,

*Corresponding author
Email addresses: salvatore.cuomo@unina.it (Salvatore Cuomo), wolfgang.erb@unipd.it (Wolfgang Erb), gsantin@fbk.eu (Gabriele Santin)

common interests, or news. Users in a social network produce a huge amount of digital information. For example, the publishing of 100 posts with photos on Facebook generates already an estimated traffic of 500MB. (see `https://www.att.com/support/data-calculator` for such estimates). For this reason, data science techniques specifically developed for social network analysis and data mining display a remarkable growth and provide promising opportunities for the inference of novel knowledge.

Several application-relevant questions related to networks such as expert finding, topic monitoring, or disease outbreak detection involve the resolution of an Influence Maximization (IM) problem, i.e., the determination of an optimal set of network elements that have, in a properly defined sense, a maximal impact on the other elements of the network. For instance, IM was addressed in [1] to find potential customers in a market by adopting a probabilistic method based on Markov Random Fields [2]. A second example is given [3], where a natural greedy strategy for the solution of a discrete stochastic optimization problem is used to study the influence of users in a social network.

From a mathematical point of view, the natural framework for social network analysis is to represent network users and mutual interactions by a *graph*, a conceptual structure where network users correspond to abstractions called *nodes* and each related pair of nodes is called *link* or *edge*.

We will study Influence Maximization on graphs using tools from Graph Signal Processing (GSP). This research field, outlined in [4, 5], provides a general mathematical framework to analyze and process data organized in graph structures. In this work, we will mainly focus on kernel-based methods on graphs. Such methods were recently studied in [6, 7] in terms of interpolation and approximation of graph signals with graph basis functions (GBFs) and partition of unity methods. In [8], an application of feature-augmented graph basis functions in a semi-supervised machine learning framework was studied. Further theoretical and computational aspects related to uncertainty principles on graphs can be found in [9].

Formally, given a graph G with a set V of nodes, a diffusion model Φ, and a positive real number N (called *budget*), the IM problem is an optimization problem that aims at finding a subset of nodes $W_N^{(\mathrm{IM})}$ such that

$$W_N^{(\mathrm{IM})} = \arg \max_{|W|=N} \Phi(W) \tag{1}$$

with $W \subseteq V$. In other words, the problem (1) consists in calculating a set $W_N^{(\mathrm{IM})}$ of N nodes such that the influence spread of $W_N^{(\mathrm{IM})}$, under the given diffusion model Φ, is maximized.

Beside a proper model for the influence spread $\Phi(W)$, two crucial computational issues are relevant in (1), : (i) the computational complexity for the calculation of the influence spread $\Phi(W)$ from the initial subset of nodes W, and (ii) the maximization of the influence spread. Both problems might be computationally expensive for large graphs, and it is well-known that (1) is in general NP-hard in complexity [10]. Strategies to overcome this issue are generally based

on greedy approaches to deal with the combinatorial source in (ii), while the computation of the influence spread is, for instance in independent cascade and linear threshold models, addressed by the usage of Monte Carlo simulations [3], and in combination with time-efficient heuristics [11, 12]. Important for the performance of the greedy algorithm are the properties of the influence function Φ. Good approximation results are guaranteed when the influence function Φ is monotone and submodular [10].

In this paper, we introduce and discuss kernel-based models for IM on graphs that are inspired by kriging techniques in the geosciences [13] and the optimal positioning of sensors, as for instance outlined in [14, 15, 16]. After some mathematical considerations on kernel-based approximation on graphs, we investigate a greedy algorithm where the function Φ is defined by the variance term of a Gaussian process regression (its square root will be referred to as power function). This Gaussian process variance depends solely on the covariance kernel of the process and can be interpreted as an information entropy related to the nodes W. In comparison to the classical formulation (1) for IM in which a spread function is maximized, our model for IM is based on the minimization of the entropy functional Φ. Further, compared to a stochastic diffusion model in which a stochastic simulation is necessary to calculate the spread, our kernel-based variance term Φ is deterministic. In the corresponding P-greedy algorithm for the approximate solution of the optimization problem a simple update scheme can be implemented in order to reduce the computational costs. For this reason, one of the main advantages of our deterministic IM model is a low computational complexity. Furthermore, our kernel-based approach offers a large flexibility in the choice of the covariance kernel, allowing to select the kernel according to a data-driven learning approach. We will present several numerical experiments to highlight the importance of this model for IM on graphs, in particular as a group centrality measure in networks.

The paper is organized as follows. Section 2 recalls the necessary tools from GSP and graph spectral theory. In particular, a connection between positive definite Graph Basis Functions (GBFs) and stationary Gaussian Processes on graphs is established. Based on these tools, Section 3 introduces the kernel-based model for IM, which is then approximated via the P-greedy algorithm presented in Section 4. Section 5 provides several numerical experiments for IM on graphs and analyses the kernel-based influence spread as a measure for group centrality and in comparison with other existing notions of influence spread on graph.

2. Gaussian process regression and kernels on graphs

2.1. Preliminaries on graph theory

In this work, we will use the following mathematical terminology to describe a graph $G = (V, E, \mathbf{L})$ and its elements:

(i) The vertex set $V = \{\mathrm{v}_1, \ldots, \mathrm{v}_n\}$ describes the n nodes $\mathrm{v}_1, \ldots, \mathrm{v}_n$ of the graph G.

(ii) The set $E \subseteq V \times V$ contains all possible edges $e_{i,i'} = (\mathrm{v}_i, \mathrm{v}_{i'}) \in E$, $i \neq i'$, of the graph G. We assume that the graph is undirected, i.e., that with $e_{i,i'} \in E$ also $e_{i',i}$ is contained in E and both describe the same edge. With these assumptions, the graph G is simple, i.e. it contains no multiple edges and no self-loops.

(iii) The strength of the connection between the vertices in V is modeled with a symmetric matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ usually referred to as graph Laplacian. We assume that $\mathbf{L}$ is a general matrix of the form (cf. [17, Section 13.9])

$$\begin{array}{ll} \mathbf{L}_{i,i'} < 0 & \text{if } i \neq i' \text{ and } \mathrm{v}_i, \mathrm{v}_{i'} \text{ are connected,} \\ \mathbf{L}_{i,i'} = 0 & \text{if } i \neq i' \text{ and } \mathrm{v}_i, \mathrm{v}_{i'} \text{ are not connected,} \\ \mathbf{L}_{i,i} \in \mathbb{R} & \text{for } i \in \{1, \ldots, n\}. \end{array} \tag{2}$$

Here, the connection weights of the edges $e_{i,i'} \in E$ are encoded in the negative non-diagonal elements $\mathbf{L}_{i,i'}$, $i \neq i'$, of $\mathbf{L}$. They are linked to the symmetric adjacency matrix $\mathbf{A}$ in the following way:

$$\mathbf{A}_{i,i'} := \begin{cases} -\mathbf{L}_{i,i'}, & \text{if } e_{i,i'} \in E, \\ 0, & \text{otherwise.} \end{cases}$$

There are several possibilities to fix the diagonal entries of $\mathbf{L}$. The most common form is the standard graph Laplacian $\mathbf{L}_S$ defined as

$$\mathbf{L}_S = \mathbf{D} - \mathbf{A} \tag{3}$$

with the diagonal degree matrix $\mathbf{D}$ given by

$$\mathbf{D}_{i,i'} := \begin{cases} \sum_{j=1}^{n} \mathbf{A}_{i,j}, & \text{if } i = i', \\ 0, & \text{otherwise.} \end{cases}$$

A major advantage of the definition (3) is that the matrix $\mathbf{L}_S$ is positive semi-definite. Related to this is also the normalized graph Laplacian $\mathbf{L}_N$ given by

$$\mathbf{L}_N = \mathbf{D}^{-1/2} \mathbf{L}_S \mathbf{D}^{-1/2}.$$

For the normalized Laplacian $\mathbf{L}_N$ all diagonal entries are equal to 1 and the spectrum is contained in the interval $[0, 2]$. More detailed information regarding combinatorial graph theory and the properties of the different graph Laplacians can be found in [18, 17].

2.2. The Fourier transform of graph signals

Information on the vertices of the graph is usually encoded in terms of graph signals $x : V \to \mathbb{R}$. As the vertices v_i, $i \in \{1, \ldots, n\}$ are ordered, we have a natural representation of a signal as a vector $x = (x(\mathrm{v}_1), \ldots, x(\mathrm{v}_n))^\intercal \in \mathbb{R}^n$. The

corresponding n-dimensional vector space $\mathcal{L}(G)$ of graph signals can therefore be regarded as an Euclidean space with the inner product

$$y^\intercal x := \sum_{i=1}^{n} x(\mathrm{v}_i) y(\mathrm{v}_i).$$

If we define the unit signals $\delta_{\mathrm{v}_{i'}} \in \mathcal{L}$ as $\delta_{\mathrm{v}_{i'}}(\mathrm{v}_i) = \delta_{i,i'}$ for $i, i' \in \{1, \dots, n\}$, the system $\{\delta_{\mathrm{v}_1}, \dots, \delta_{\mathrm{v}_n}\}$ forms a canonical orthonormal basis of $\mathcal{L}(G)$.

The Fourier transform on graphs is a natural extension of related Fourier concepts in classical harmonic analysis and provides an indispensable tool for the analysis and processing of graph signals. It is defined by the eigendecomposition of the graph Laplacian (the most common choices for the graph Fourier transform are $\mathbf{L}_S$ or $\mathbf{L}_N$)

$$\mathbf{L} = \mathbf{U}\mathbf{M}_\lambda \mathbf{U}^\intercal.$$

where $\mathbf{M}_\lambda = \mathrm{diag}(\lambda) = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ is the diagonal matrix containing the increasingly ordered eigenvalues λ_i, $i \in \{1, \dots, n\}$, of $\mathbf{L}$ as diagonal entries, and $\mathbf{U}$ is an orthogonal matrix containing the corresponding orthonormal eigenvectors of $\mathbf{L}$ as columns. The system of eigenvectors $\hat{G} = \{u_1, \dots, u_n\}$ forms an orthonormal basis of the signal space $\mathcal{L}(G)$. Given the Fourier basis $\hat{G}$, the graph Fourier transform $\hat{x}$ of a signal x, and the respective inverse Fourier transform are defined as

$$\hat{x} := \mathbf{U}^\intercal x = (u_1^\intercal x, \dots, u_n^\intercal x)^\intercal, \quad \text{and} \quad x = \mathbf{U}\hat{x}.$$

In analogy to classical Fourier analysis, the basis elements u_k, $k \in \{1, \dots, n\}$, can be regarded as Fourier modes of the graph and the entries $\hat{x}_k = u_k^\intercal x$ describe exactly the frequency content of the signal x with respect to the Fourier mode u_k.

As a general graph contains no inherent group structure, it is not possible to define the translation of a graph signal directly on the graph. The graph Fourier transform allows to bypass this problem and to introduce a generalized form of translation on graphs. This is done in terms of a convolution operator $\mathbf{C}_y$ acting on a signal x as

$$\mathbf{C}_y x = \mathbf{U}\mathbf{M}_{\hat{y}}\mathbf{U}^\intercal x. \tag{4}$$

Then, the convolution $\mathbf{C}_{\delta_\mathrm{v}} x$ of a signal x with the unit signal δ_v can be interpreted as a generalized translate of the signal x on the graph G by the node v. Indeed, if G is endowed with a group structure and the Fourier basis $\hat{G}$ corresponds to the classical set of characters of the group G, then $\mathbf{C}_{\delta_\mathrm{v}} x = x(\cdot - \mathrm{v})$ is precisely a shift of the signal x by the group element v. More information about different applications of the graph Fourier transform, for instance, as analysis, filtering or decomposition tool, can, for instance, be found in [9, 4, 19, 5].

2.3. Positive definite Graph Basis Functions (GBFs)

Graph basis functions (GBFs) are simple and efficient tools for kernel-based interpolation and approximation methods on graphs [6, 8]. In this GBF-method,

the approximation spaces are built upon generalized shifts of a GBF f such that every signal x is approximated by a linear combination

$$x_*(\mathrm{v}) = \sum_{i=1}^{N} c_i \mathbf{C}_{\delta_{\mathrm{w}_i}} f(\mathrm{v})$$

of generalized shifts $\mathbf{C}_{\delta_{\mathrm{w}_i}} f$. The coefficients c_i are calculated based on the information of x on a sampling set $W = \{\mathrm{w}_1, \ldots, \mathrm{w}_N\} \subset V$. In this sense, the idea of GBFs is closely related to well-known theories in the literature on approximation with radial basis functions (RBFs)in the Euclidean space [20, 21], or spherical basis functions on the unit sphere.

An important prerequisite for the uniqueness of the approximant x_* is the positive definiteness of the GBF f. A signal $f \in \mathcal{L}(G)$ is called positive definite (see [9]) if the kernel matrix

$$\mathbf{K}_f = \begin{pmatrix} \mathbf{C}_{\delta_{\mathrm{v}_1}} f(\mathrm{v}_1) & \mathbf{C}_{\delta_{\mathrm{v}_2}} f(\mathrm{v}_1) & \ldots & \mathbf{C}_{\delta_{\mathrm{v}_n}} f(\mathrm{v}_1) \\ \mathbf{C}_{\delta_{\mathrm{v}_1}} f(\mathrm{v}_2) & \mathbf{C}_{\delta_{\mathrm{v}_2}} f(\mathrm{v}_2) & \ldots & \mathbf{C}_{\delta_{\mathrm{v}_n}} f(\mathrm{v}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{\delta_{\mathrm{v}_1}} f(\mathrm{v}_n) & \mathbf{C}_{\delta_{\mathrm{v}_2}} f(\mathrm{v}_n) & \ldots & \mathbf{C}_{\delta_{\mathrm{v}_n}} f(\mathrm{v}_n) \end{pmatrix}$$

is positive definite. It was shown in [6] that a signal f is positive definite if and only if $\hat{f}_k > 0$ for all $k \in \{1, \ldots, n\}$. Further, the kernel function K_f given by $K_f(\mathrm{v}, \mathrm{w}) := \mathbf{C}_{\delta_{\mathrm{w}}} f(\mathrm{v})$ has the Mercer decomposition

$$K_f(\mathrm{v}, \mathrm{w}) = \mathbf{C}_{\delta_{\mathrm{w}}} f(\mathrm{v}) = \sum_{k=1}^{n} \hat{f}_k \, u_k(\mathrm{v}) \, u_k(\mathrm{w}).$$

In this way, a positive definite GBF f induces, in a natural way, an inner product $\langle x, y \rangle_{K_f}$ and a norm $\|x\|_{K_f}$ by

$$\langle x, y \rangle_{K_f} = \sum_{k=1}^{n} \frac{\hat{x}_k \hat{y}_k}{\hat{f}_k} = \hat{y}^{\intercal} \mathbf{M}_{1/\hat{f}} \hat{x} \quad \text{and} \quad \|x\|_{K_f} = \sqrt{\sum_{k=1}^{n} \frac{\hat{x}_k^2}{\hat{f}_k}}.$$

The space $\mathcal{L}(G)$ of signals endowed with this inner product is a reproducing kernel Hilbert space $\mathcal{N}_{K_f}$ with the reproducing kernel given as K_f.

2.4. Stationary Gaussian Processes on Graphs

We assume now that $X = (X(\mathrm{v}_1), \ldots, X(\mathrm{v}_n))$ is a non-degenerate Gaussian random process on the graph G, i.e. that X is a n-variate Gaussian random variable with a positive definite covariance matrix. We further assume that X is stationary in the following sense:

Definition 1. *A Gaussian random process X on the graph G is called (wide-sense) stationary if the following two conditions are satisfied [22]:*

(i) The expectation value $\mathrm{E}(X(\mathrm{v}_i)) = \mu$ is constant for all nodes v_i of G.

(ii) The covariance matrix of the Gaussian process is invariant under the generalized shift operator $\mathbf{C}_{\delta_{\mathrm{v}_i}}$ and given by the positive definite matrix

$$\mathrm{E}\left((X(\mathrm{v}_i)-\mu)(X(\mathrm{v}_j)-\mu)\right) = \mathbf{K}_{f,i,j} = \mathbf{C}_{\delta_{\mathrm{v}_i}} f(\mathrm{v}_j).$$

In particular, every stationary Gaussian random process X is uniquely determined by the mean value $\mu \in \mathbb{R}$ and a positive definite GBF f providing the covariance matrix $\mathbf{K}_f$ of the process. Further, the density function φ_X of the stationary Gaussian process X is given by

$$\varphi_X(x) = \frac{\exp(-\frac{1}{2}(x-\mu\mathbb{1})^{\intercal}\mathbf{K}_f^{-1}(x-\mu\mathbb{1}))}{\sqrt{(2\pi)^n|\mathbf{K}_f|}},$$

where $\mathbb{1} = (1,\dots,1)^{\intercal} \in \mathbb{R}^n$ and $|\mathbf{K}_f|$ denotes the determinant of $\mathbf{K}_f$.

2.5. Gaussian process regression (kriging)

Gaussian processes provide simple stochastic models for data regression. We consider additive white noise ϵ on the graph G as a stationary Gaussian process with zero mean and covariance matrix $\sigma^2\mathbf{I}_n$. Then, if X denotes a stationary Gaussian process on G with mean $\mu = 0$ and covariance matrix $\mathbf{K}_f$ the random signal

$$Y = X + \epsilon \tag{5}$$

is also a stationary Gaussian process with mean $\mu = 0$ and covariance matrix $\mathbf{K}_f + \sigma^2\mathbf{I}_n$. From N known samples $Y(\mathrm{w}_i) = y_i$, $i \in \{1,\dots,N\}$, on a given sampling set $W = \{\mathrm{w}_1,\dots,\mathrm{w}_N\}$ our goal is to find a predictor x_* of the process X based on the given sampling information and the model in (5). As X and Y are both Gaussian, the $N+1$ dimensional random vector $(Y(W), X(\mathrm{v})) = (Y(\mathrm{w}_1),\dots,Y(\mathrm{w}_N), X(\mathrm{v}))$ is also Gaussian for any node $\mathrm{v} \in V$. The corresponding normal distribution is given as

$$\begin{pmatrix} Y(W) \\ X(\mathrm{v}) \end{pmatrix} \sim N\left(\mathbf{0}, \begin{pmatrix} \mathbf{K}_{f,W} + \sigma^2\mathbf{I}_n & \mathbf{C}_{\delta_W} f(\mathrm{v}) \\ \mathbf{C}_{\delta_W} f(\mathrm{v})^{\intercal} & \mathbf{K}_{f,\mathrm{v}} \end{pmatrix}\right),$$

in which the covariance matrix is built upon $\mathbf{K}_{f,\mathrm{v}} = \mathbf{C}_{\delta_{\mathrm{v}}} f(\mathrm{v})$, the positive definite matrix $\mathbf{K}_{f,W} \in \mathbb{R}^{N\times N}$, and the vector $\mathbf{C}_{\delta_W} f(\mathrm{v}) \in \mathbb{R}^N$ given as

$$\mathbf{K}_{f,W} = \begin{pmatrix} \mathbf{C}_{\delta_{\mathrm{w}_1}} f(\mathrm{w}_1) & \dots & \mathbf{C}_{\delta_{\mathrm{w}_N}} f(\mathrm{w}_1) \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{\delta_{\mathrm{w}_N}} f(\mathrm{w}_1) & \dots & \mathbf{C}_{\delta_{\mathrm{w}_N}} f(\mathrm{w}_N) \end{pmatrix}, \tag{6}$$

and $\mathbf{C}_{\delta_W} f(\mathrm{v}) = \left(\mathbf{C}_{\delta_{\mathrm{w}_1}} f(\mathrm{v}),\cdots,\mathbf{C}_{\delta_{\mathrm{w}_N}} f(\mathrm{v})\right)^{\intercal}$. For this, also the conditional distribution of the random variable $X(\mathrm{v})$ given the information $Y(\mathrm{w}_i) = y_i$,

$i \in \{1, \dots, N\}$ is a Gaussian distribution with the (conditional) expectation value

$$\begin{aligned} x_*(\mathrm{v}) &= \mathrm{E}(X(\mathrm{v})|\ Y(\mathrm{w}_1) = y_1, \dots, Y(\mathrm{w}_N) = y_N) \\ &= \sum_{i=1}^{N} c_i \mathbf{C}_{\delta_{\mathrm{w}_i}} f(\mathrm{v}), \end{aligned} \tag{7}$$

and the coefficients $(c_1, \dots, c_N)^\intercal$ calculated as (see [23])

$$\begin{pmatrix} \mathrm{c}_1 \\ \vdots \\ c_N \end{pmatrix} = \left(\mathbf{K}_{f,W} + \sigma^2 \mathbf{I}_N\right)^{-1} \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}. \tag{8}$$

Furthermore, the standard deviation of this conditional Gaussian distribution is given as

$$\mathrm{P}_W(\mathrm{v}) = \left(\mathbf{K}_{f,\mathrm{v}} - \mathbf{C}_{\delta_W} f(\mathrm{v})^\intercal (\mathbf{K}_{f,W} + \sigma^2 \mathbf{I}_n)^{-1} \mathbf{C}_{\delta_W} f(\mathrm{v})\right)^{1/2}. \tag{9}$$

This posterior standard deviation $\mathrm{P}_W(\mathrm{v})$, and, in the same way, the squared variance, depend on the node set W but not on the sampling values $y_1, \dots, y_N$. It is a measure of uncertainty for the process X on a node v if (arbitrary) information is provided on the sampling set W. In the literature on scattered data approximation the quantity $\mathrm{P}_W(\mathrm{v})$ is also referred to as *p*ower function [24, 21].

3. Modelling influence maximization on graphs using Gaussian process variance minimization

Our model to determine a set $W_N^{(\mathrm{IM})} \subset V$ of N graph nodes with a maximal influence on the graph G is based on the solution of the following optimization problem:

$$W_N^{(\mathrm{IM})} = \arg \min_{|W|=N} \underbrace{\max_{\mathrm{v} \in V} |\mathrm{P}_W(\mathrm{v})|}_{\Phi(W)}. \tag{10}$$

In comparison to the classical formulation for IM on graphs provided in (1), our model is based on the minimization of the functional Φ which is determined by the power function P_W. In the following, we analyze the properties of this IM model and interpret it in terms of the minimization of an information entropy. Observe that P_W, and thus $W_N^{(\mathrm{IM})}$, has an hidden dependence on the noise variance σ^2 (see (9)). Since our aim is to select influential nodes independently from a possible presence of noise, we set from now on $\sigma := 0$.

Property 1. *The model* (10) *for IM has the properties:*

(*i*) *The model function is bounded by*

$$0 \leq \Phi(W) \leq \max_{\mathrm{v} \in V} \sqrt{\mathbf{K}_{f,\mathrm{v}}} \quad \textit{for all } W \subset V.$$

(*ii*) *The model function is monotonic (see [24]), i.e.,*

$$\Phi(W_{N+1}^{(\mathrm{IM})}) \leq \Phi(W_N^{(\mathrm{IM})}).$$

This model contains a rather large set of parameters that have to be determined a priori or extracted from given data. The main components are

- In the construction of the graph G from a set of given nodes and edges, the Laplacian $\mathbf{L}$ is not always a priori fixed. The Laplacian $\mathbf{L}$ contains all the connection weights representing the local influence of the nodes to their neighboring nodes.
- Once the Laplacian $\mathbf{L}$ is determined, the choice of the kernel $\mathbf{K}_f$ determines the global spread of information on the graph G. Such a kernel can be given a priori or calculated from data by a learning approach. With respect to the last point, an approach to kernel selection based on the training and inference via sampling random Fourier features has been proposed for kernels on $\mathbb{R}^d$ in [25], and a variational improvement is given in [26]. Additionally, methods based on the maximization of a likelihood function are very popular in machine learning (see e.g. [27], Chapter 14). Two popular kernels for a priori choices are given in Example 1.

Example 1. *Two well-known examples of positive definite GBF-kernels* $\mathbf{K}_f$ *on a graph* G *are the following [6]:*

(*i*) *The diffusion kernel on a graph [28] is given by*

$$\mathbf{K}_{f_{e^{-t\mathbf{L}}}} = e^{-t\mathbf{L}} = \sum_{k=1}^{n} e^{-t\lambda_k} u_k u_k^{\intercal},$$

where λ_k, $k \in \{1, \ldots, n\}$, *denote the eigenvalues of the Laplacian* $\mathbf{L}$. *This matrix is positive definite for all* $t \in \mathbb{R}$. *The coefficients* $e^{-t\lambda_k}$ *in the given Mercer decomposition are precisely the Fourier coefficients of* $f_{e^{-t\mathbf{L}}}$.

(*ii*) *The variational spline*

$$\mathbf{K}_{f_{(\epsilon \mathbf{I}_n + \mathbf{L})^{-s}}} = (\epsilon \mathbf{I}_n + \mathbf{L})^{-s} = \sum_{k=1}^{n} \frac{1}{(\epsilon + \lambda_k)^s} u_k u_k^{\intercal}$$

is positive definite for $\epsilon > -\lambda_1$ *and* $s > 0$. *These type of GBF-kernels were investigated in [29, 30] in terms of signal interpolation. Interpolants based on the translates of this GBF minimize an energy functional and are referred to as variational or polyharmonic splines.*

When leaving the graph setting for a short moment, similar models as the one given in (10) can be found in a multitude of applications related to geosciences or in approximation theory. This provides us with a better understanding on how the model can be interpreted in the graph setting.

The model (10) **in geostatistical kriging.** In geostatistics, methods related to the Gaussian process regression described in Section 2.5 are usually referred to as kriging methods [13]. In kriging, the square of the power function P_W is called the kriging variance and its global minimization is used for the quantitative design of infill sampling strategies in the investigation of geological sites. In particular, optimization problems as the one in (10) are widely used criteria for the location of new sampling positions [31, 32] or the optimal placement of sensors [14, 15, 16]. In the latter articles, a modification of the model (10) is investigated in which a mutual information quantity between the nodes W and the remaining nodes $V \setminus W$ is maximized. In our numerical experiments, we will also take this modification into account and compare it with the proposed model (10).

The model (10) **in approximation theory:** The power function plays a central role in the error estimation of kernel-based interpolation in approximation theory. The notion has first been introduced in [24], and later used in a multitude of works. We refer to Chapter 11 in [21] for a comprehensive treatment of this topic. Minimization criteria as in (10) in the context of approximation theory are discussed in the next section.

Interpretation of the model (10) **for IM on graphs:** Classical models as the independent cascade (IC) or the linear threshold (LT) model [3] describe IM on graphs trough the propagation of information, where, conceptually similar to the spread of a disease, an infected node is transmitting information to its neighbors with a given probability. On the other side, the model (10) can be rather considered as a group centrality on graphs in which $\Phi(W)$ corresponds to a maximal information entropy of the node set W. The minimization problem in (10) can be interpreted as the search for a set of nodes that minimizes the maximal entropy (or information uncertainty) on the graph among all sets of a given size.

4. P-greedy for variance minimization

The determination of the set $W_N^{(\mathrm{IM})} \subset V$ from the exact solution of (10) requires to consider all the $\binom{n}{N} \geq (n/N)^N$ possible subsets of N elements among the n nodes of the graph. This combinatorial problem is practically intractable already for moderate-sized graphs.

To overcome this limitation, we replace the exact criterion (10) with an approximation based on a very simple, yet effective greedy procedure. Namely, starting from the empty set of nodes $W_0^{(\mathrm{IM})} = \emptyset$, at each iteration $k = 1, \ldots, N$

we update it by adding a new node as

$$\begin{aligned} &\mathrm{w}_k \in \arg\max_{\mathrm{w}\in V\setminus W_{k-1}^{(\mathrm{IM})}} \mathrm{P}_{W_{k-1}^{(\mathrm{IM})}}(w), \\ &W_k^{(\mathrm{IM})} := W_{k-1}^{(\mathrm{IM})} \cup \{\mathrm{w}_k\}, \end{aligned} \tag{11}$$

where the inclusion relation "$\in$" denotes the fact that the node realizing the maximum may be non-unique. Moreover, (9) implies that $\mathrm{P}_{W_0^{(\mathrm{IM})}}(w) = \sqrt{\mathbf{K}_{f,\mathrm{w}}}$ for all $\mathrm{w} \in V$, and thus the selection is well defined for all k. Finally, observe that the method can also be run to enrich an initial set of nodes, i.e., one may start with $W_0^{(\mathrm{IM})} = W_0$, with any $W_0 \subset V$.
The heuristic (11) for the approximate solution of (10) is well-known in spatial statistics [13, 33], and has been introduced in [34] under the name of P-greedy algorithm in the context of kernel approximation on $\mathbb{R}^d$. Despite its simplicity, the method has been proven to be asymptotically optimal (see [35, 36]), i.e., greedily selected nodes provide the same rate of decay of $\mathrm{P}_{W_N^{(\mathrm{IM})}}$ as optimally chosen ones.

These asymptotic properties seem to be valid, in a limited sense, also in the present graph setting, although the asymptotic limit $N \to \infty$ is not applicable on a finite graph. Nevertheless, the excellent performance of the method that will be observed in Section 5 seems to suggest that a similar quasi-optimality may be proven here. We leave the investigation of this point open for future study.

4.1. Computational complexity

The P-greedy algorithm can be implemented very efficiently, see [37, 38]. Indeed, an efficient formula is available to update $\mathrm{P}_{W_{k-1}^{(\mathrm{IM})}}$ to $\mathrm{P}_{W_k^{(\mathrm{IM})}}$, and thus (11) requires only the update of the solution of a k-dimensional triangular linear system and a simple linear search over $V \setminus W_{k-1}^{(\mathrm{IM})}$ at each iteration.

In the present graph setting the computational aspects remain the same and, in particular, the same implementation of the method is seamlessly extendable for kernels on graphs. Assuming that a single evaluation of a shifted GBF at a node requires constant time, the overall complexity to obtain a P-greedy solution $W_k^{(\mathrm{IM})}$ is of order $\mathcal{O}(nk^2)$ [37]. If no further heuristics are used the P-greedy approach to obtain the most influencing nodes of a graph has therefore a computational advantage over costly Monte-Carlo simulations as required in the independent cascade model, or over the minimization of the mutual information quantity as presented in [15].

We remark that the P-greedy algorithm has been extended to the case of matrix-valued kernels in [39]. In our setting, this would allow to approximate vector-valued signals over the graph.

5. Numerical experiments

In this section we test our method under different points of view. In Section 5.1 we set some parameters and configurations to clarify the setup of the fol-

lowing experiments. Section 5.2 is devoted to an in-depth data-driven tuning of the algorithm in order to obtain an optimized method, which is then compared with state of the art alternatives in Section 5.3.

5.1. Experimental setup

All the numerical tests are realized with a Python implementation of the method discussed in the previous sections, which builds on the Python library NetworkX [40] for graph representation. The implementation, including the code to replicate the experiments, is made public at [41].

We use as test cases some small- to medium-sized graphs that are described in Table 1, and that can be generated by the code in [41]. For visualization convenience only, all these graphs have a two-dimensional position attribute associated with each node.

Table 1: Graphs used in the numerical experiments

ID	N. nodes	N. edges	Description
`bunny`	1035	9468	The nodes are obtained by projecting into the plane the points of the *Stanford bunny* [42], and thinning the points with a separation radius of 0.0025. The nodes that are within a radius of 0.01 are then linked.
`sensor1`	79	210	This sensor graph consisting of 79 randomly chosen points in the plane has been generated for this work.
`airports`	1311	13169	Airline transportation network from data collected in 2019 by the US Bureau of Transportation Statistics, as processed in [43]. Each node is an airport, and edges represent connections with weights given by the number of passenger who travelled in any of the two directions. The airports outside the longitude-latitude box $[-180^o, 70^o] \times [15^o, 75^o]$ are discarded.

The influence maximization algorithm of Section 4 is run on the entire set of nodes (i.e., each node may potentially be chosen). We remark that whenever multiple nodes maximize the greedy criterion, the one with the largest degree is selected. The constant signal with value 1 on all nodes is used to validate the algorithm, namely residuals are computed with respect to the approximation of this signal. The algorithm is terminated when a maximal number of nodes are

selected, or when either the maximal value of the squared standard deviation, or the maximal valued of the residual fall below the tolerance $\tau := 10^{-12}$. This ensures a certain stability in the algorithm (see [38]). As GBFs we use the diffusion and the variational spline kernels (see Example 1), with parameters that are defined in each experiment.

As an example, we visualize in Figure 1 the first 20 nodes selected by the algorithm on `bunny` (first row, left panel), `sensor1` (first row, right panel), and `airports` (second row), using a diffusion kernel with parameter $t = -10$.

Remark 1. *It is worth noticing that in the first case the nodes appear to be spatially uniform, as one may expect since the graph has uniformly distributed nodes, and this is in line with the behavior of the node selection algorithm when applied in $\mathbb{R}^d$ (see [36]). On the other hand, also in the second case a kind of uniformity seems to be present, even if constrained to the graph's topology. The investigation of the properties of these node distributions is an interesting topic that we intend to address in the future.*

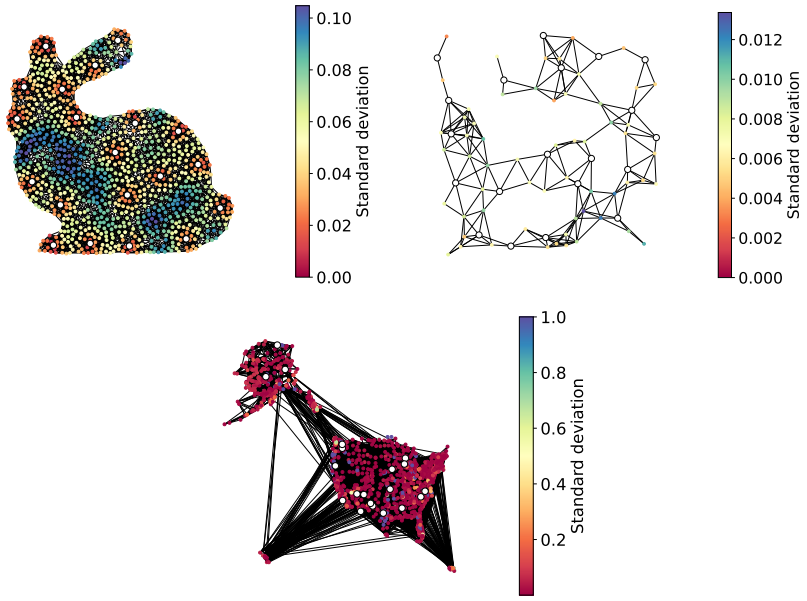

Figure 1: Example of the first 20 points (circled nodes) selected by the new algorithm using a diffusion kernel on `bunny` (first row, left), `sensor1` (first row, right), and `airports` (second row). The nodes are colored according to the value of the standard deviation $\mathrm{P}_{W_{20}^{(\mathrm{IM})}}$.

5.2. Data-driven model learning

Since our definition of influence is depending on the chosen kernel (including its parameters), it is of upmost relevance to devise a systematic methodology to learn the kernel from data.

As an initial attempt in this direction, we use here a cross validation approach. Namely, we use standard k-fold cross validation to optimize the kernel and its parameters in order to minimize the residual in the approximation of the signal with constant value 1. This target signal is chosen as it equally weights all nodes in the graph, and thus its approximation seems to be a suitable proxy to model a uniform spread of the information.

Remark 2. *We remark that k-fold cross validation has a computational cost that scales linearly with respect to $k \cdot N_p$, where $N_p \in \mathbb{N}$ is the number of tested parameter configurations. Indeed, for each parameter configuration the method trains k models, each on a different splitting of the dataset, where the cost of the training of the single model is exactly the one described in Section 4.1.*

It is clear that the bottleneck of this approach is the size of the parameter grid that has to be tested, and to address this issue there exist randomized sampling strategies that avoid testing each single configuration and instead sample a fixed and possibly small number of parameter values, with the goal of having an approximated optimization with reduced cost.

Both deterministic and randomized cross validation have the advantage of not requiring a smooth dependence of the model on the parameters, so that, as mentioned, one can treat for example the kernel as an optimizable parameter. Even more importantly, gradient-based optimization methods such as Marginal Likelihood Estimation (see e.g. Section 14.3 in [27]) require the computation of the gradient of a functional of the model with respect to the parameters, which is difficult to obtain for methods as the present one, where the number and set of approximation nodes $W_M^{(\mathrm{IM})}$ are themselves dependent on the parameters.

To be more specific, on `bunny` we compare a variational spline GBF with parameters $\epsilon = 0.01$, $s = -1$, and a diffusion GBF with $t = -10$, with their optimized variants. The optimization is realized with 5-fold cross validation over the parameter grids described in Table 2. As a model selector we use the mean error in the approximation of the constant signal over the entire graph. The selected parameters are reported also in Table 2. We stress that it would be easily possible to include the kernel itself as an optimizable parameter, but here we prefer to derive two different models, one per kernel, in order to compare them.

Table 2: Range of the search grid for the parameter optimization (first row) and selected parameters (second row) for the variational spline and the diffusion GBFs. Each parameter range is discretized into 25 logarithmically equally spaced values.

	variational spline		diffusion
	ϵ	s	t
search interval	$[10^{-16}, 10^{0}]$	$[-10^{1}, -10^{-1}]$	$[-10^{2}, -10^{-2}]$
optimized	10^{-6}	-2.15	-31.62

The first 20 points resulting from the four selections are depicted in Figure

2 and colored with the absolute value of the final standard deviation. Observe that this quantity is kernel-dependent, and thus different scales are present in the plots. It is immediately clear, especially for the variational spline, that optimized parameters yield better distributed nodes and thus possibly a more credible notion of influence maximization. This is even more evident and quantitatively assessed in Figure 3, where we report the decay of the maximal standard deviation on the entire graph as a function of the selected nodes. In this case, to facilitate the comparison of the four configurations, we show standard deviations normalized by their initial maximum. To further quantify the different results, we also report in Table 3 the numerical values of the standard deviation for the representative values $N = 5, 10, 15, 20$. The decay of the maxima is much faster for the optimized kernels, with significantly smaller values for the variational spline.

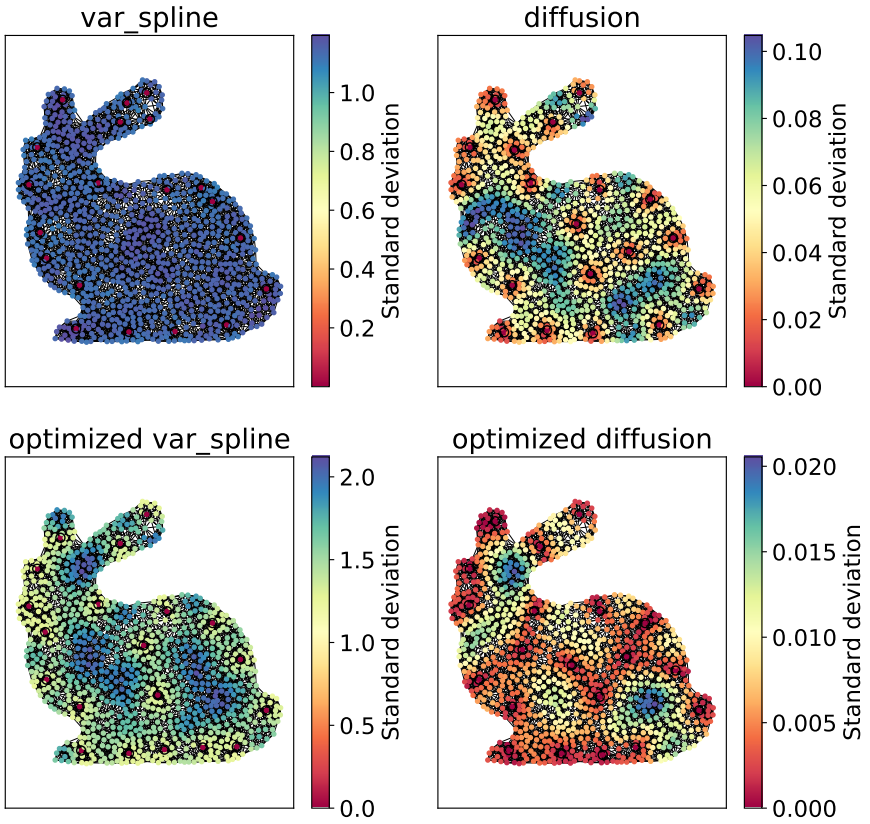

Figure 2: First 20 nodes selected by the new algorithm on `bunny` using the variational spline GBF (first column) or the diffusion GBF (second column), before (first row) and after the optimization of their parameters (second row).

Table 4 additionally reports the ids of the first selected nodes. Observe that

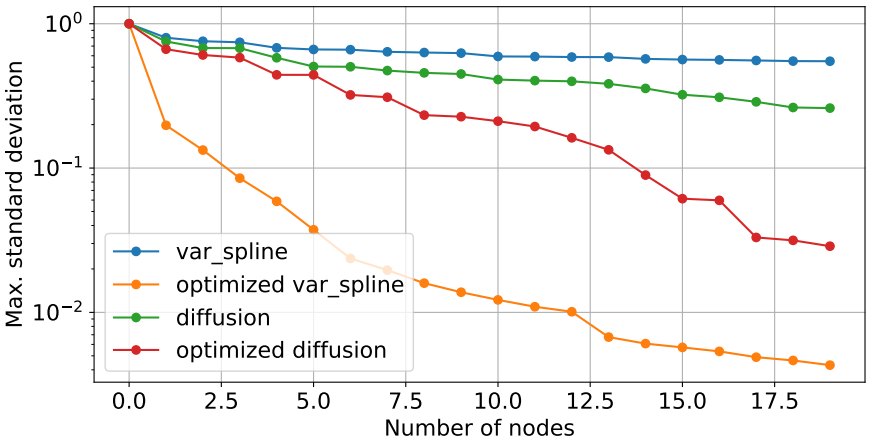

Figure 3: Decay of the standard deviation as a function of the number of the nodes selected by the new algorithm on `bunny` using the variational spline GBF or the diffusion GBF, before and after the optimization of their parameters. All standard deviations are normalized to an initial value of 1. See also Table 3.

Table 3: Representative values of the standard deviation as a function of the number of the nodes selected by the new algorithm on `bunny` using the variational spline GBF or the diffusion GBF, before and after the optimization of their parameters. All standard deviations are normalized to an initial value of 1. See also Figure 3.

	variational spline		diffusion	
N	not optimized	optimized	not optimized	optimized
5	6.81e-01	5.88e-02	5.81e-01	4.43e-01
10	6.26e-01	1.38e-02	4.49e-01	2.27e-01
15	5.71e-01	6.08e-03	3.56e-01	8.94e-02
20	5.50e-01	4.31e-03	2.60e-01	2.88e-02

there is a certain stability for the different configurations, and this is a good indication of the quality of the selection process. On the other hand, notice that slight changes in the order have already an effect in the decay of the standard deviation (see Figure 3), and this points to the importance of a proper kernel selection to maximize the performances of the method. We remark in particular that the same method may lead to significanly different results depending on the specific choice of the kernel and its parameters. A sistematic parameter optimization procedure, as discussed in this section, is thus a key ingredient to obtain a reliable node selection.

Table 4: First 5 nodes selected by the new algorithm on `bunny` using the variational spline GBF or the diffusion GBF, either optimized or not.

variational spline		diffusion	
not optimized	optimized	not optimized	optimized
4	4	4	4
730	344	730	730
164	164	164	776
776	17	775	164
459	919	121	793

5.3. Comparison with state of the art methods

We compare now on `sensor1` the quality of the nodes selected by our method to other state of the art techniques, namely Pagerank (PR) [44], Independent Cascade (IC) [45], and the mutual information-based sensor placement method for Gaussian Processes introduced in [15] (MI). For the first algorithm we use the implementation provided by NetworkX, while for the second we use an own implementation that is made available in [41]. The same code is used also for MI. Indeed, using the language and notation of this paper, MI can be implemented as a greedy method where the new point is selected by maximizing a ratio between two power functions, namely (see Algorithm 1 in [15])

$$\mathrm{w}_k \in \arg\max_{\mathrm{w}\in V\setminus W_{k-1}^{(\mathrm{IM})}} \frac{\mathrm{P}^2_{W_{k-1}^{(\mathrm{IM})}}(w)}{\mathrm{P}^2_{V\setminus\left(W_{k-1}^{(\mathrm{IM})}\cup\{w\}\right)}(w)}.$$

We remark that the same paper introduces also efficient approximations of the same selection criterion (see Section 5 in [15]), but we are content with this version as we do not aim at comparing the computational efficiency of the two methods, but rather the selected nodes.

The algorithm IC is run with a spread probability $p = 0.2$ and with 500 independent runs at each iteration, and furthermore it is applied in a greedy mode, i.e., the algorithm is run sequentially to select each time the next most influential node. For our method we use also in this case the variational spline

GBF, with parameters optimized in the same way as in the previous section, resulting into $\epsilon = -4.64 \cdot 10^{-12}$, $s = -1$ (the parameters are different from the ones of the previous section because the underlying graph is different). We use the same values also to run MI, in order to facilitate the comparison of the two kernel-based greedy algorithms.

The first 10 nodes selected by each algorithm are shown in Figure 4 and listed in Table 5, where we additionally report the 10 nodes of higher degree. In this case, the three nodes $1, 23, 64$ are selected by all the four algorithms and are among the 10 nodes of highest degree. Moreover, the new algorithm shares 5 of the first 10 nodes with PR, while IC and PR share 3. Remarkably, the new algorithm selects the first 10 nodes in the exact same way of MI, while maximizing a significantly simpler selection rule which does not require to employ approximation steps. The two algorithms are not identical though: in Figure 4 we visualize the percentage of selected nodes that are chosen by both algorithms, for each $n = 1, \dots, |V|$. Observe that the selected nodes remain the same up to the 17 th, where the two algorithms start to make a different selection up to reaching a minimum of 80.7% of shared nodes.

To quantify the quality of the selected nodes, we compare them according to the two metrics defined by IC and by our algorithm. Namely, for each incremental set of nodes we compute the max value of the standard deviation over the entire graph using the optimized variational spline GBF, and the IC score, i.e., the fraction of nodes not reached by IC starting from the set of given nodes. The decay of these metrics are shown in the first row of Figure 6. As expected, both the current method and IC minimize the respective score, but it is interesting to observe that the new method performs almost optimally also with respect to the IC metric. If one instead considers the mean standard deviation (second row of Figure 6), it is remarkable that PR outperforms both methods, even if slightly. Observe also that the new method and MI perfectly overlaps in all the three panels of Figure 6. This is an indication that, even if the two lists of nodes are not equal (only the first 17 of the 20 nodes coincide), their performances are very similar. To better quantify the comparison, in Table 6 we report the values of these metrics for $N = 5, 10, 15, 20$.

Table 5: First 10 nodes selected by the new algorithm, IC, pagerank, and MI when applied to `sensor1`, and 10 nodes of highest degree. The nodes that are selected by all algorithms are in boldface fonts, while the ones selected both by PR and by the new algorithm are underlined.

kernel	**64**	**1**	**23**	8	6	43	74	16	65	73
IC	**23**	**64**	**1**	0	58	**16**	**14**	22	**17**	76
PR	**1**	6	56	**23**	51	**64**	71	24	30	43
MI	**64**	**1**	**23**	8	6	43	74	**16**	65	73
node degree	**23**	**64**	71	**1**	24	35	42	53	60	6

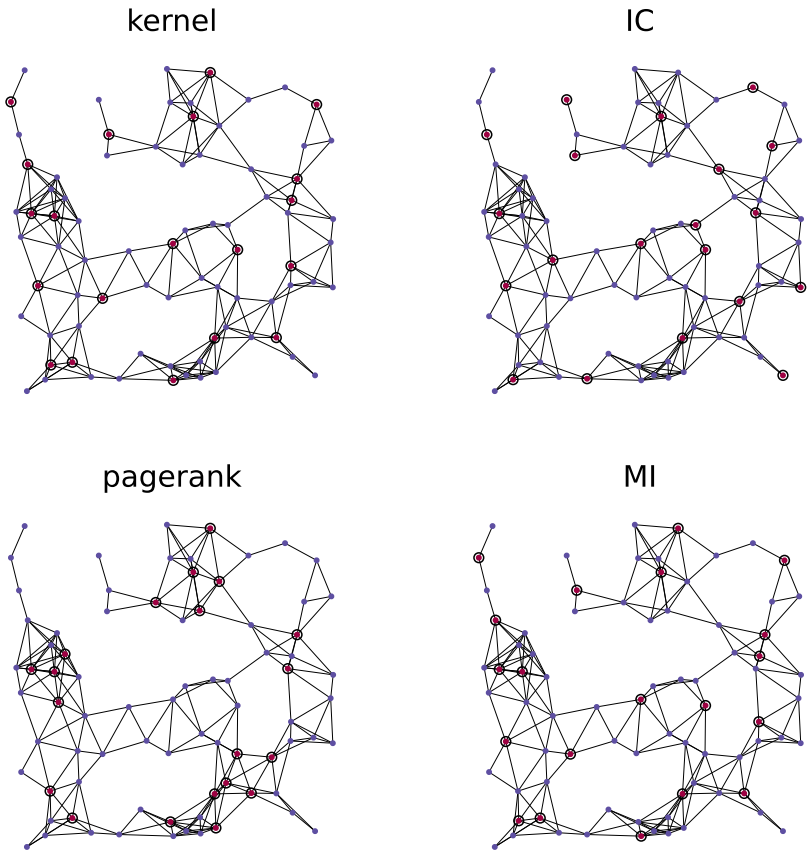

Figure 4: Distribution of the first 10 nodes selected by the new algorithm (top left), IC (top right), pagerank (bottom left), and IM (bottom right) when applied to `sensor1`.

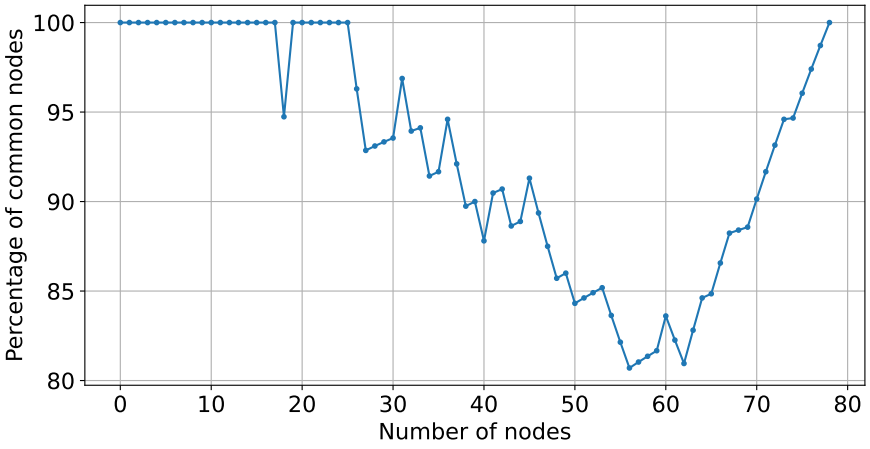

Figure 5: Percentage of the nodes that are in common between MI and the new algorithm, as a function of the number of selected nodes $n = 1, \ldots, |V|$, when applied to `sensor1`.

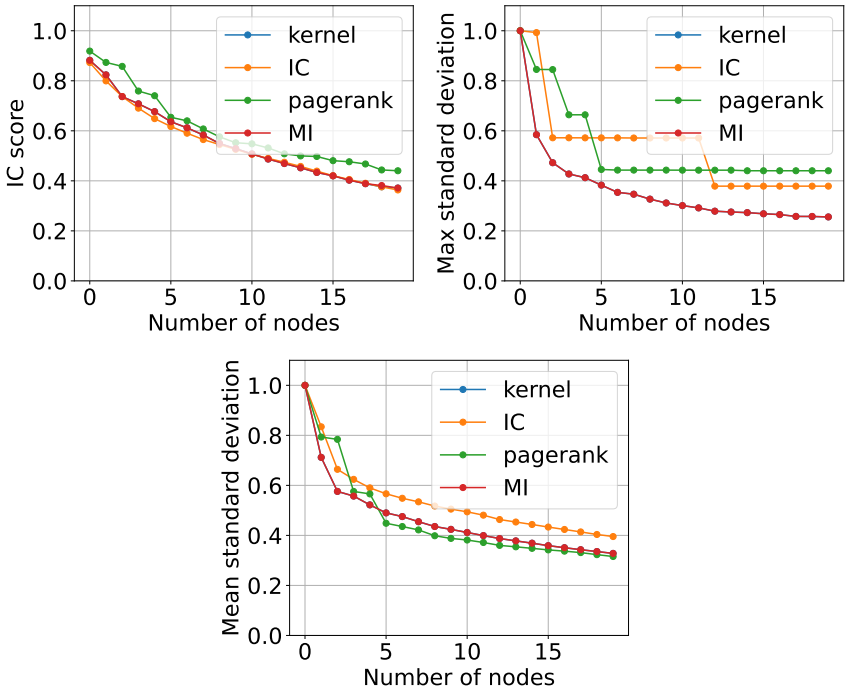

Figure 6: Decay of the IC score (first line left), of the max standard deviation (first line right), of the mean standard deviation (second line) as a function of the number of nodes selected by the new algorithm, IC, pagerank, and MI when applied to `sensor1`. Observe that the curves for the new method and for MI perfectly overlap. See also Table 6.

Table 6: Values of the different metrics discussed in Section 5.3 as a function of the number of nodes selected by the new algorithm, IC, pagerank, when applied to `sensor1`. See also Figure 6.

	IC score			
N	kernel	IC	PR	MI
5	6.77e-01	6.49e-01	7.40e-01	6.77e-01
10	5.29e-01	5.26e-01	5.52e-01	5.29e-01
15	4.34e-01	4.39e-01	4.97e-01	4.34e-01
20	3.71e-01	3.64e-01	4.40e-01	3.71e-01

	Max std			
N	kernel	IC	PR	MI
5	4.13e-01	5.72e-01	6.64e-01	4.13e-01
10	3.11e-01	5.71e-01	4.43e-01	3.11e-01
15	2.73e-01	3.79e-01	4.40e-01	2.73e-01
20	2.56e-01	3.79e-01	4.40e-01	2.56e-01

	Mean std			
N	kernel	IC	PR	MI
5	5.22e-01	5.90e-01	5.66e-01	5.22e-01
10	4.24e-01	5.05e-01	3.88e-01	4.24e-01
15	3.69e-01	4.44e-01	3.48e-01	3.69e-01
20	3.28e-01	3.95e-01	3.16e-01	3.28e-01

5.4. Centrality-based influence maximization

The maximal kriging variance in the model (10), i.e., the maximal information entropy of a set W as discussed in Section 3, can be interpreted as a group centrality on the graph describing the impact of a small set of nodes W to the entire graph. It makes therefore sense to compare the centrality given by the spread function $\Phi(W)$ and associated to the power function of a GBF-kernel to other well known group centralities as, for instance, group degree, group closeness or group betweeness [46].

To introduce these metrics we denote as $\Sigma(u, v)$ the set of shortest paths between two nodes $u, v \in V$ and, given an arbitrary $\sigma \in \Sigma(u, v)$, we write $\mathrm{dist}(u, v) := \mathrm{len}(\sigma)$ for the path distance between two nodes. This can be extended in the usual way to a node-to-set distance, i.e., $\mathrm{dist}(v, W) := \min_{w \in W} \mathrm{dist}(v, w)$ if $v \in V$ and $W \subset V$.

The group degree centrality of W is the fraction of nodes outside W which are connected to W, i.e.,

$$C_{\mathrm{deg}}(W) := \frac{|\{v \in V \setminus W : \exists w \in W \text{ s.t. } (v, w) \in E\}|}{|V \setminus W|}.$$

The group betweenness centrality of W is the fraction of shortest paths between nodes outside W that pass through W, i.e.,

$$C_{\mathrm{betw}}(W) := \sum_{u,v \in V \setminus W} \frac{|\{\sigma \in \Sigma(u, v) : \exists w \in W \cap \sigma\}|}{|\Sigma(u, v)|}.$$

The group closeness centrality of W is the reciprocal of the average distance from W of the nodes in $V \setminus W$, i.e.,

$$C_{\mathrm{clos}}(W) := \frac{|V \setminus W|}{\sum_{v \in V \setminus W} \mathrm{dist}(v, W)}.$$

Roughly speaking, C_{clos} and C_{deg} measure how well $V \setminus W$ can be reached from V, while C_{betw} quantifies the centrality of W as an hub to move between nodes in V.

We use these metrics to quantify the centrality of the lists of nodes selected on the graph `sensor1` by IC, pagerank, and by the new algorithm, as described in the previous section. We omit MI, since the selected nodes coincide with the ones of the new algorithm. Figure 7 reports the evolution of these metrics for the three algorithms. It is clear that the new algorithm is very effective: after an initial phase where IC is superior, it reaches the maximal centrality in fewer iterations for C_{deg} and C_{clos}.

6. Conclusion

For influence maximization on graphs, we presented and investigated a kernel-based approach for the description of information spread on graphs. The proposed deterministic approach is based on GBF-kernels of graphs and the minimization of a variance term related to Gaussian process regression. In order to

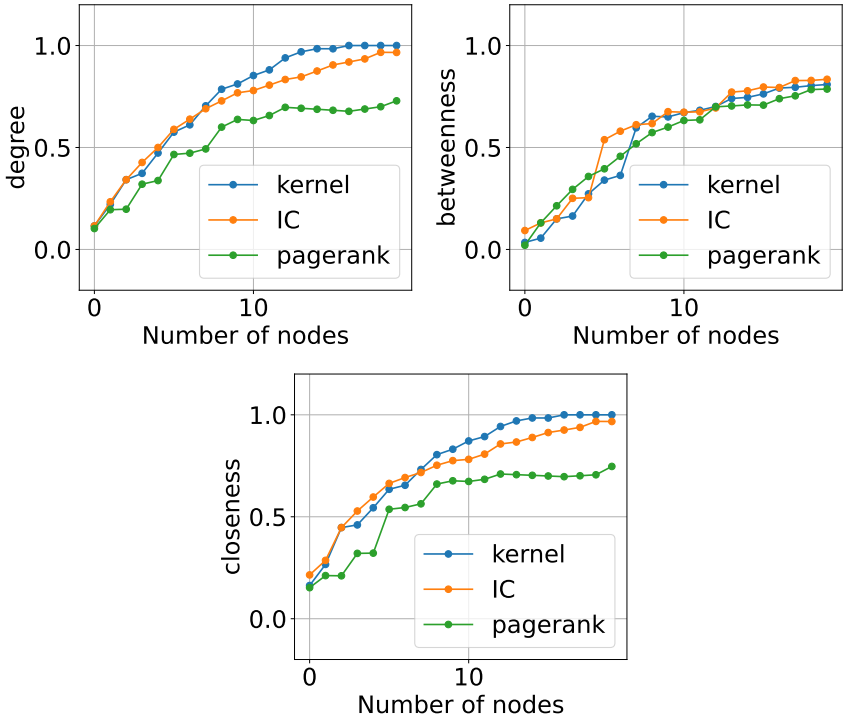

Figure 7: Increase of the group centrality of the nodes selected by IC, pagerank, and the new algorithm on the graph `sensor1`. The three panels show the centrality metrics C_{deg}, C_{betw}, C_{clos} (from left to right).

select the influential nodes of a graph, we implemented a cost-efficient approximate minimization of the variance by an efficient P-greedy algorithm. We could experimentally show that our scheme is more efficient than respective greedy schemes involving stochastic spread models, for instance, the independent cascade model. An emerging issue for our model is the proper selection of the covariance kernel which has a direct impact on the performance of the method. To overcome this problem, we tested a data-driven approach to determine the kernel and used machine learning techniques for the parameter tuning of the model. Challenging questions for future research include smart strategies for a data-driven extraction of the kernel, machine learning methodologies for an optimal selection of the model parameters, as well as theoretical questions on possible distributions of the selected nodes.

Acknowledgments

This work was partially supported by INdAM-GNCS. This research has been accomplished within the two Italian research groups on approximation theory RITA and UMI T.A.A.

References

[1] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 57–66.

[2] H. Rue and L. Held, *Gaussian Markov random fields: theory and applications*. CRC press, 2005.

[3] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 137–146.

[4] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[5] L. Stanković, M. Daković, and E. Sejdić, "Introduction to graph signal processing," in *Vertex-Frequency Analysis of Graph Signals*. Springer, 2019, pp. 3–108.

[6] W. Erb, "Graph signal interpolation with positive definite graph basis functions," *Applied and Computational Harmonic Analysis*, vol. 60, pp. 368–395, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1063520322000276

[7] R. Cavoretto, A. De Rossi, and W. Erb, "Partition of unity methods for signal processing on graphs," *J. Fourier Anal. Appl.*, vol. 27, p. Art. 66, 2021.

[8] W. Erb, "Semi-supervised learning on graphs with feature-augmented graph basis functions," *arXiv preprint arXiv:2003.07646*, 2020.

[9] ——, "Shapes of uncertainty in spectral graph theory," *IEEE Transactions on Information Theory*, vol. 67, no. 2, pp. 1291–1307, 2021.

[10] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[11] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 1029–1038.

[12] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, 2014, pp. 946–957.

[13] N. A. Cressie, *Statistics for Spatial Data*, 2nd ed. John Wiley and Sons, 1993.

[14] W. F. Caselton and J. V. Zidek, "Optimal monitoring network designs," *Statistics and Probability Letters*, vol. 2, no. 4, pp. 223–227, 1984.

[15] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. 2, 2008.

[16] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection," *Journal of Machine Learning Research*, vol. 9, no. 12, 2008.

[17] C. Godsil and G. F. Royle, *Algebraic graph theory*. Springer Science & Business Media, 2001, vol. 207.

[18] F. R. Chung and F. C. Graham, *Spectral graph theory*, ser. CBMS Regional Conference Series in Mathematics. American Mathematical Soc., 1997, no. 92.

[19] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, 2016.

[20] R. Schaback and H. Wendland, "Approximation by positive definite kernels," in *Advanced Problems in Constructive Approximation*, ser. International Series in Numerical Mathematics, M. Buhmann and D. Mache, Eds., vol. 142, 2002, pp. 203–221.

[21] H. Wendland, *Scattered Data Approximation*, ser. Cambridge Monographs on Applied and Computational Mathematics. Cambridge: Cambridge University Press, 2005, vol. 17.

[22] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3462–3477, 2017.

[23] G. Wahba, *Spline models for observational data*. SIAM, 1990, vol. 59.

[24] R. Schaback, "Error estimates and condition numbers for radial basis function interpolation," *Adv. Comput. Math.*, vol. 3, no. 3, pp. 251–264, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF02432002

[25] C.-L. Li, W.-C. Chang, Y. Mroueh, Y. Yang, and B. Póczos, "Implicit kernel learning," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 2007–2016.

[26] X. Zhen, H. Sun, Y. Du, J. Xu, Y. Yin, L. Shao, and C. Snoek, "Learning to learn kernels with variational random features," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 409–11 419.

[27] G. E. Fasshauer and M. McCourt, *Kernel-Based Approximation Methods Using MATLAB*, ser. Interdisciplinary Mathematical Sciences. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2015, vol. 19.

[28] R. I. Kondor and J. D. Lafferty, "Diffusion kernels on graphs and other discrete input spaces," in *Proceedings of the Nineteenth International Conference on Machine Learning*, ser. ICML '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, p. 315–322.

[29] I. Pesenson, "Variational splines and paley–wiener spaces on combinatorial graphs," *Constructive Approximation*, vol. 29, no. 1, pp. 1–21, 2009.

[30] J. P. Ward, F. J. Narcowich, and J. D. Ward, "Interpolating splines on graphs for data science applications," *Applied and Computational Harmonic Analysis*, vol. 49, no. 2, pp. 540–557, 2020.

[31] R. L. Bras and I. Rodriguez-Iturbe, *Random functions and hydrology*. Reading (Mass.) : Addison-Wesley, 1985.

[32] D. E. Scheck and D.-R. Chou, "Optimum locations for exploratory drill holes," *International Journal of Mining Engineering*, vol. 1, pp. 343–355, 1983.

[33] M. McKay, R. Beckman, and W. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[34] S. De Marchi, R. Schaback, and H. Wendland, "Near-optimal data-independent point locations for radial basis function interpolation," *Adv. Comput. Math.*, vol. 23, no. 3, pp. 317–330, 2005. [Online]. Available: http://dx.doi.org/10.1007/s10444-004-1829-1

[35] G. Santin and B. Haasdonk, "Convergence rate of the data-independent P-greedy algorithm in kernel-based approximation," *Dolomites Res. Notes Approx.*, vol. 10, pp. 68–78, 2017. [Online]. Available: www.emis.de/journals/DRNA/9-2.html

[36] T. Wenzel, G. Santin, and B. Haasdonk, "A novel class of stabilized greedy kernel approximation algorithms: Convergence, stability and uniform point distribution," *Journal of Approximation Theory*, vol. 262, p. 105508, 2021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021904520301441

[37] M. Pazouki and R. Schaback, "Bases for kernel-based spaces," *J. Comput. Appl. Math.*, vol. 236, no. 4, pp. 575–588, 2011. [Online]. Available: http://www.ams.org/mathscinet-getitem?mr=2843040

[38] G. Santin and B. Haasdonk, "Kernel methods for surrogate modeling," in *Volume 1 System- and Data-Driven Methods and Algorithms*, P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, Eds. De Gruyter, 2021, pp. 311–354. [Online]. Available: https://doi.org/10.1515/9783110498967-009

[39] D. Wittwar and B. Haasdonk, "Greedy algorithms for matrix-valued kernels," in *Numerical Mathematics and Advanced Applications ENUMATH 2017*, F. A. Radu, K. Kumar, I. Berre, J. M. Nordbotten, and I. S. Pop, Eds. Cham: Springer International Publishing, 2019, pp. 113–121.

[40] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11 – 15.

[41] S. Cuomo, W. Erb, and G. Santin, "GBF, Python implementation," 2021. [Online]. Available: https://github.com/GabrieleSantin/GraphBasisFunction

[42] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 303–312. [Online]. Available: https://doi.org/10.1145/237170.237269

[43] D. for Science, "Graphs4Sci," 2021. [Online]. Available: https://github.com/DataForScience/Graphs4Sci

[44] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[45] G. D'Angelo, L. Severini, and Y. Velaj, "Influence maximization in the independent cascade model." in *ICTCS*. Citeseer, 2016, pp. 269–274.

[46] M. G. Everett and S. P. Borgatti, "The centrality of groups and classes," *The Journal of Mathematical Sociology*, vol. 23, no. 3, pp. 181–201, 1999. [Online]. Available: https://doi.org/10.1080/0022250X.1999.9990219