Active Semi-Supervised Learning Using Sampling Theory for Graph Signals

Akshay Gadde, Aamir Anis and Antonio Ortega
Department of Electrical Engineering
University of Southern California, Los Angeles
agadde@usc.edu, aanis@usc.edu, ortega@sipi.usc.edu

ABSTRACT

We consider the problem of offline, pool-based active semi-supervised learning on graphs. This problem is important when the labeled data is scarce and expensive whereas unlabeled data is easily available. The data points are represented by the vertices of an undirected graph with the similarity between them captured by the edge weights. Given a target number of nodes to label, the goal is to choose those nodes that are most informative and then predict the unknown labels. We propose a novel framework for this problem based on our recent results on sampling theory for graph signals. A graph signal is a real-valued function defined on each node of the graph. A notion of frequency for such signals can be defined using the spectrum of the graph Laplacian matrix. The sampling theory for graph signals aims to extend the traditional Nyquist-Shannon sampling theory by allowing us to identify the class of graph signals that can be reconstructed from their values on a subset of vertices. This approach allows us to define a criterion for active learning based on sampling set selection which aims at maximizing the frequency of the signals that can be reconstructed from their samples on the set. Experiments show the effectiveness of our method.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

Keywords

Active semi-supervised learning; Graph signal processing; Sampling theory; Graph signal filtering

KDD'14, August 24–27, 2014, New York, NY, USA.

ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623760.

1. INTRODUCTION

In many real-life machine learning tasks, labeled data is scarce whereas unlabeled data is easily available. Active semi-supervised learning is an effective approach for such scenarios. A semi-supervised learning technique must not only learn from the labeled data but also from the inherent clustering present in the unlabeled data [29]. Further, when the labeling is expensive, it is better to let the learner choose the data points to be labeled so that it can pick the most informative and representative labels. Thus, in an active learning scenario, the goal is to achieve the maximum gain in terms of learning ability for a given, and small, number of label queries. In this paper, we propose a novel approach to active semi-supervised learning based on recent advances in sampling theory for graph signals.

Active learning has been studied in different problem scenarios such as online stream-based sampling, adaptive sampling etc. (see [23] for a review). We focus on the problem of pool-based batch-mode active semi-supervised learning, where there is a large static collection of unlabeled data from which a very small percentage of data points have to be selected in order to be labeled. Batch operation (i.e., selecting a *set* of data points to be labeled) is more realistic in scenarios such as crowdsourcing where it would not be practical to submit for labeling one data point at a time. Further, in this paper we focus on the problem of optimizing batches of any size without using any label information, which would be the case when selecting the first batch of data points to be labeled. We leave for future work the problem of incorporating labeled data, which would allow labels obtained for the first batch to be used to optimize data point selection for the second batch, and so on.

Applying a graph perspective to semi-supervised learning is not new. In a graph-based formulation, the data points are represented by nodes of a graph and the edges capture the similarity between the nodes they connect. For example, the weight on an edge might be a function of the distance between the two points in the feature space chosen for the classification task. The membership function of a given class can be thought of as a "graph signal", which has a scalar value at each of the nodes (e.g., 1 or 0 depending on whether or not the data point belongs to the class). Since features have been chosen to be meaningful for the classification task, it is reasonable to expect that nodes that are close together in the feature space will be likely to have the same label. Conversely, nodes that are far away in the feature space are less likely to have the same label. Thus, we expect the membership function to be *smooth* on the graph, i.e., moving

from a node to its neighbors in the graph is unlikely to lead to changes in the membership. Thus, the semi-supervised learning problem can be viewed as a problem of interpolating a smooth graph signal. This view has led to many effective techniques such as MinCut [4], Gaussian random fields and harmonic functions [30], local and global consistency [28], manifold regularization [3] and spectral graph kernels [25].

Active learning has also benefited from this graph based-view. Many active learning approaches use the graph to quantify the quality of sampling sets [7, 9, 10]. One methodology is to try and pick a subset of nodes which captures the underlying low-dimensional manifold represented by the graph. Another is to pick the nodes to be labeled in such a way that unlabeled nodes are strongly connected to them. Some methods pick those samples which lead to minimization of generalization error bound. We discuss some of these methods in Section 4.

Many of the semi-supervised methods mentioned above are *global*, in the sense that they require inversion or eigen-decomposition of large matrices associated with the underlying graph. This poses a problem in scalable and distributed implementation of these algorithms. Most graph-based active learning methods suffer from the same problem. Another issue with these methods is that they do not give conditions under which the graph signal can be uniquely and perfectly interpolated from its samples on the chosen subset.

In recent years, there has been a significant amount of work devoted to the study of graph signal processing. The focus of this work has been to extend to the context of graph signals, theoretical results and tools that are well established in the context of conventional signal processing [24]. In particular, there have been contributions to the design of graph wavelets [11], graph filterbanks [17], etc. A key challenge in graph signal processing is to design *localized* algorithms that scale well with graph sizes, i.e., the output at each vertex should only depend on its local neighborhood.

In this paper we leverage our recent work on graph signal sampling and interpolation [18, 1]. We show that the newly developed theoretical results provide a rigorous and unified framework to select points to be labeled and subsequently perform semi-supervised learning. Our framework provides conditions under which a graph signal can be uniquely recovered from its values on a subset of vertices. These conditions lead to a powerful greedy algorithm for choosing the best nodes for labeling. The proposed algorithm is well motivated through a compelling graph theoretic interpretation. We give a numerically efficient way to implement the proposed algorithm which makes it scalable. We also give an effective and efficient semi-supervised learning method that is closely tied to the label selection algorithm and is theoretically well-justified. Both our algorithms are well-suited for a large-scale distributed implementation. We show that our method outperforms several state of the art methods by testing on multiple real datasets.

The rest of the paper is organized as follows. Section 2 reviews our recent work on sampling theory for graph signals. In Section 3 we apply the framework of sampling theory to derive the proposed active semi-supervised learning approach. Section 4 summarizes the related prior work. Experiments are presented in Section 5. Finally, we provide some concluding remarks in Section 6.

2. SAMPLING THEORY FOR GRAPH SIGNALS

We begin by briefly describing the theory of sampling for graph signals formulated in our previous work [18, 1].

2.1 Notation

Throughout this paper, we consider simple, connected, undirected, and weighted graphs $G = (\mathcal{V}, E)$ with nodes numbered from the set $\mathcal{V} = \{1, 2, \ldots, N\}$, and edges $E = \{(i, j, w_{ij})\}, i, j \in \mathcal{V}$, where (i, j, w_{ij}) denotes an edge of weight w_{ij} between nodes i and j, with $w_{ii} = 0$. In the present context, the weights denote similarity between the respective nodes. The degree d_i of a node i is defined as the sum of the weights of edges connected to node i, and the degree matrix of the graph is a diagonal matrix defined as $\mathbf{D} = \text{diag}\{d_1, d_2, \ldots, d_N\}$. The adjacency matrix $\mathbf{W}$ of the graph is an $N \times N$ matrix with $\mathbf{W}_{ij} = w_{ij}$ and the combinatorial Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. We shall use the symmetric normalized form of the adjacency and the Laplacian matrices defined as $\boldsymbol{\mathcal{W}} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ and $\boldsymbol{\mathcal{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ respectively. $\boldsymbol{\mathcal{L}}$ is a symmetric positive semi-definite matrix and has a set of real eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N \leq 2$ and a corresponding orthogonal set of eigenvectors denoted as $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}$. A subset of nodes of the graph is denoted as a collection of indices $\mathcal{S} \subset \mathcal{V}$, with $\mathcal{S}^c = \mathcal{V} \setminus \mathcal{S}$ denoting its complement set. A restriction of a matrix $\mathbf{A}$ to rows in set $\mathcal{S}_1$ and columns in set $\mathcal{S}_2$ is denoted by the submatrix $\mathbf{A}_{\mathcal{S}_1,\mathcal{S}_2}$ and for the sake of brevity $\mathbf{A}_{\mathcal{S},\mathcal{S}} = \mathbf{A}_{\mathcal{S}}$. Also, $\mathbf{0}$ and $\mathbf{1}$ denote all-zeroes and all-ones vectors of appropriate sizes.

A graph signal is defined as a scalar-valued discrete mapping $f : V \to \mathbb{R}$, such that $f(i)$ is the value of the signal on node i. For ease of notation, it can also be represented as a vector $\mathbf{f} \in \mathbb{R}^N$ with indices corresponding to the node indices in the graph. In this paper, the signals of interest will be the membership functions associated with the various labels of interest in the classification problem. *Sampling* a graph signal $\mathbf{f}$ onto a subset of nodes $\mathcal{S}$, known as the *sampling set*, is realized by retaining the signal's values on the nodes in $\mathcal{S}$. The sampled signal is denoted by $\mathbf{f}(\mathcal{S})$, which is a vector of reduced length $|\mathcal{S}|$. In our context, a sampled graph signal will include the membership information for the data points that have been labeled.

2.2 Preliminaries

The classical Nyquist-Shannon sampling theorem establishes an upper limit on the bandwidth of signals that can be uniquely reconstructed when sampled at a given sampling rate. To have an analogous result in the realm of graphs, one needs a notion of frequency for graph signals. Such a spectral interpretation is provided by the eigenvalues and eigenvectors of the Laplacian matrix $\boldsymbol{\mathcal{L}}$, similar to the Fourier transform in traditional signal processing. The eigenvalues can be thought of as frequencies and indicate the variation in the eigenvectors: a high eigenvalue implies higher variation in the corresponding eigenvector [24]. Since the eigenvectors are orthogonal, they form a basis in $\mathbb{R}^N$. Thus, the *Graph Fourier Transform* (GFT) of a signal $\mathbf{f}$ is defined as its projection onto the eigenvectors of the graph Laplacian, i.e. $\tilde{\mathbf{f}}(\lambda_i) = \langle \mathbf{f}, \mathbf{u}_i \rangle$, or more compactly, $\tilde{\mathbf{f}} = \mathbf{U}^T\mathbf{f}$.

In this context, a smooth or low-pass graph signal can be obtained by forcing high frequency GFT coefficients to

vanish. More formally, an *ω-bandlimited signal* on a graph is defined to have zero GFT coefficients for frequencies above its bandwidth ω, i.e. its spectral support is restricted to the set of frequencies $[0, \omega]$. The space of all ω-bandlimited signals is known as the *Paley-Wiener* space and is denoted by $PW_\omega(G)$ [20]. Note that $PW_\omega(G)$ is a subspace of $\mathbb{R}^N$.

With the notion of frequency introduced via the GFT, one can frame an adequate *sampling theory* for graph signals using the following ingredients:

- **P1:** *Cutoff frequency* - For a given subset of nodes $\mathcal{S}$, find the cut-off frequency ω, such that any $\mathbf{f} \in PW_\omega(G)$ can be exactly recovered from its samples $\mathbf{f}(\mathcal{S})$.
- **P2:** *Optimal sampling set* - For a given cut-off frequency ω, find the the smallest subset of nodes $\mathcal{S}_{\text{opt}}$ (i.e. with minimum $|\mathcal{S}_{\text{opt}}|$) such that all signals $\mathbf{f} \in PW_\omega(G)$ can be uniquely recovered from their samples $\mathbf{f}(\mathcal{S}_{\text{opt}})$ on $\mathcal{S}_{\text{opt}}$.
- **P3:** *Reconstruction algorithm* - Given samples $\mathbf{f}(\mathcal{S})$ of a graph signal $\mathbf{f}$ on a subset of nodes $\mathcal{S}$, find the reconstructed signal values $\mathbf{f}(\mathcal{S}^c)$ on the complementary subset $\mathcal{S}^c$.

Note that for regular sampling in the traditional signal processing, problems **P1** and **P2** are reciprocal, i.e., knowing one automatically leads to the solution of the other. However, this does not hold for irregular sampling, as in the case of graph signals. Next, we briefly describe the solution to each of the problems above, and refer to [18, 1] for the details.

2.3 P1: Cut-off frequency

Let $L_2(\mathcal{S}^c)$ denote the space of all graph signals that are zero everywhere except possibly on the nodes in $\mathcal{S}^c$, i.e., $\forall \phi \in L_2(\mathcal{S}^c), \phi(\mathcal{S}) = 0$. Also, let $\omega(\phi)$ denote the bandwidth of a graph signal ϕ, i.e., the value of the maximum non-zero frequency of that signal. Then the following theorem can be proved [1]:

THEOREM 1 (SAMPLING THEOREM). *For a graph G, with normalized Laplacian $\boldsymbol{\mathcal{L}}$, any signal $\mathbf{f} \in PW_\omega(G)$ can be perfectly recovered from its values on a subset of nodes $\mathcal{S} \subset \mathcal{V}$ if and only if*

$$\omega < \omega_c(\mathcal{S}) \triangleq \inf_{\phi \in L_2(\mathcal{S}^c)} \omega(\phi) \tag{1}$$

where $\omega_c(\mathcal{S})$ is the cut-off frequency.

The theorem leads to a cut-off frequency that is *lower* than the minimum bandwidth of any signal in $L_2(\mathcal{S}^c)$. Intuitively, a signal $\phi \in L_2(\mathcal{S}^c)$ can be added to any input signal $\mathbf{f}$ without affecting its sampled version (since ϕ is identically zero for all vertices that are sampled, i.e., those in $\mathcal{S}$). Thus, if there existed a $\phi \in L_2(\mathcal{S}^c)$ such that $\phi \in PW_\omega(G)$ we would have that both $\mathbf{f}$ and $\phi + \mathbf{f}$ belong to $PW_\omega(G)$ *and* lead to the same set of samples on $\mathcal{S}$. So clearly it would not be possible to recover them both, and thus sampling of such signals in $PW_\omega(G)$ would not be possible. The condition in Theorem 1 ensures that $PW_\omega(G) \cap L_2(\mathcal{S}^c) = \{0\}$ and thus no such ϕ exists.

From Theorem 1, finding the maximum cut-off frequency for a set $\mathcal{S}$ requires finding the bandwidth $\omega(\phi^*)$ of the smoothest possible signal $\phi^* \in L_2(\mathcal{S}^c)$. A brute-force approach to this would entail computing the GFT of all signals in $L_2(\mathcal{S})$ and exhaustively searching for ϕ^*. We instead devise a computationally efficient way to approximate the bandwidth of any signal ϕ for a given integer parameter $k > 0$ as follows:

$$\omega_k(\phi) = \left(\frac{\phi^t \boldsymbol{\mathcal{L}}^k \phi}{\phi^t \phi} \right)^{1/k} \tag{2}$$

We then replace $\omega(\phi)$ in Theorem 1 by $\omega_k(\phi)$ in the objective function to obtain our estimated bandwidth:

$$\Omega_k(\mathcal{S}) = \inf_{\phi \in L_2(\mathcal{S}^c)} \omega_k(\phi) = \inf_{\phi \in L_2(\mathcal{S}^c)} \left(\frac{\phi^t \boldsymbol{\mathcal{L}}^k \phi}{\phi^t \phi} \right)^{1/k}. \tag{3}$$

Then, the smoothest possible signal ϕ^* in $L_2(\mathcal{S}^c)$ can be approximated by the minimizer ϕ_k^* in (3). Numerically, $\Omega_k(\mathcal{S})$ and ϕ_k^* can be determined from the smallest eigen-pair $(\sigma_{1,k}, \psi_{1,k})$ of the reduced matrix $(\boldsymbol{\mathcal{L}}^k)_{\mathcal{S}^c}$:

$$\Omega_k(\mathcal{S}) = \sigma_{1,k}, \tag{4}$$

$$\phi_k^*(\mathcal{S}^c) = \psi_{1,k},\ \phi_k^*(\mathcal{S}) = \mathbf{0}. \tag{5}$$

This approach does not require complete eigen-decomposition of $\boldsymbol{\mathcal{L}}$ and is computationally tractable. One can show that k controls the accuracy of the cut-off estimate (refer to [1] for details). As we increase the value of k, $\Omega_k(\mathcal{S})$ tends to give a better estimate of the cut-off frequency. Thus, there is a trade-off between accuracy of the estimate on the one hand, and complexity and numerical stability on the other that arise due to the power k in $\boldsymbol{\mathcal{L}}^k$. Moreover, $\Omega_k(\mathcal{S})$ can be proven to be always less than the actual cut-off $\omega_c(\mathcal{S})$, i.e. the Sampling Theorem still holds for the subset $\mathcal{S}$ except that the class of recoverable signals is determined to be narrower as a penalty for the cut-off approximation.

2.4 P2: Sampling set

We now describe the framework for the converse question: given a cut-off frequency ω_c for $PW_\omega(G)$, what is the smallest sampling set $\mathcal{S}_{\text{opt}}$ so that a signal $\mathbf{f} \in PW_\omega(G)$ is uniquely represented by $\mathbf{f}(\mathcal{S}_{\text{opt}})$. If K_c represents the number of eigenvalues of $\boldsymbol{\mathcal{L}}$ below ω_c, then by dimensionality considerations $|\mathcal{S}_{\text{opt}}| \geq K_c$. Also, note that $\mathcal{S}_{\text{opt}}$ may not be unique. Formally, one can use Theorem 1 and relax the true cut-off $\omega_c(\mathcal{S})$ by $\Omega_k(\mathcal{S})$, then $\mathcal{S}_{\text{opt}}$ can be found from the following optimization problem:

$$\underset{\mathcal{S}}{\text{Minimize}}\ |\mathcal{S}| \quad \text{subject to} \quad \Omega_k(\mathcal{S}) \geq \omega_c \tag{6}$$

This is a combinatorial problem because we need to compute $\Omega_k(\mathcal{S})$ for every possible subset $\mathcal{S}$.

However, this problem can be solved using a greedy heuristic to get an estimate $\mathcal{S}_{\text{est}}$ of the optimal sampling set. Starting with an empty sampling set $\mathcal{S}$ (with corresponding $\Omega_k(\mathcal{S}) = 0$) we keep adding nodes to $\mathcal{S}$ (from $\mathcal{S}^c$) one-by-one while trying to ensure maximum increase in $\Omega_k(\mathcal{S})$ at each step. The hope is that $\Omega_k(\mathcal{S})$ reaches the target cut-off ω_c with minimum number of node additions to $\mathcal{S}$. To understand which nodes should be included in $\mathcal{S}$, we introduce a binary relaxation of our cut-off formulation by defining the following matrix

$$\mathbf{M}_k^\alpha(\mathbf{t}) \triangleq \boldsymbol{\mathcal{L}}^k + \alpha\, \boldsymbol{\mathcal{D}}(\mathbf{t}), \quad k \in \mathbb{Z}^+, \alpha > 0, \mathbf{t} \in \mathbb{R}^N \tag{7}$$

where $\boldsymbol{\mathcal{D}}(\mathbf{t})$ is a diagonal matrix with $\mathbf{t}$ on its diagonal. Let $(\lambda_k^\alpha(\mathbf{t}), \mathbf{x}_k^\alpha(\mathbf{t}))$ denote the smallest eigen-pair of $\mathbf{M}_k^\alpha(\mathbf{t})$. Then, if $\mathbf{1}_\mathcal{S} : \mathcal{V} \to \{0, 1\}$ denotes the indicator function for

the subset $\mathcal{S}$ (i.e. $\mathbf{1}(\mathcal{S}) = \mathbf{1}$ and $\mathbf{1}(\mathcal{S}^c) = \mathbf{0}$), one has

$$\lambda_k^\alpha(\mathbf{1}_\mathcal{S}) = \inf_{\mathbf{x}} \left(\frac{\mathbf{x}^t \boldsymbol{\mathcal{L}}^k \mathbf{x}}{\mathbf{x}^t \mathbf{x}} + \alpha \frac{\mathbf{x}(\mathcal{S})^t \mathbf{x}(\mathcal{S})}{\mathbf{x}^t \mathbf{x}} \right) \tag{8}$$

Note that the right hand side of the equation above is simply an *unconstrained regularization* of the constrained optimization problem in (3). When $\alpha \gg 1$, the components $\mathbf{x}(\mathcal{S})$ are highly penalized during minimization. Thus, if $\mathbf{x}_k^\alpha(\mathbf{1}_\mathcal{S})$ is the minimizer in (8), then $[\mathbf{x}_\alpha^k(\mathbf{1}_\mathcal{S})](\mathcal{S}) \to \mathbf{0}$, i.e. the values on nodes $\mathcal{S}$ tend to be very small. Therefore, for $\alpha \gg 1$, we have

$$\phi_k^* \approx \mathbf{x}_k^\alpha(\mathbf{1}_\mathcal{S}), \quad (\Omega_k(\mathcal{S}))^k \approx \lambda_k^\alpha(\mathbf{1}_\mathcal{S}) \tag{9}$$

From the above equation, we observe that the problem of greedily maximizing $\Omega_k(\mathcal{S})$ is equivalent to maximizing $\lambda_k^\alpha(\mathbf{1}_\mathcal{S})$, and thus, we simply need to study the variation of $\lambda_k^\alpha(\mathbf{t})$ with $\mathbf{t}$, a real-valued vector in $\mathbb{R}^N$, at $\mathbf{t} = \mathbf{1}_\mathcal{S}$. This relaxation circumvents the combinatorial nature of our problem and has been used earlier to study graph partitioning based on Dirichlet eigenvalues [19]. The gradient of $\lambda_k^\alpha(\mathbf{t})$ with respect to $\mathbf{t}(i)$ is given by

$$\left.\frac{d\lambda_\alpha^k(\mathbf{t})}{d\mathbf{t}(i)}\right|_{\mathbf{t}=\mathbf{1}_\mathcal{S}} = \alpha \left([\mathbf{x}_\alpha^k(\mathbf{1}_\mathcal{S})](i) \right)^2 \approx \alpha(\boldsymbol{\phi}_k^*(i))^2. \tag{10}$$

This equation forms the basis of our greedy heuristic: starting with an empty $\mathcal{S}$ (i.e., $\mathbf{1}_\mathcal{S} = \mathbf{0}$), if at each step, we include the node on which the smoothest signal $\phi_k^* \in L_2(\mathcal{S}^c)$ has maximum energy (i.e., $\mathbf{1}_\mathcal{S}(i) \leftarrow 1, i = \arg\max_j \left[(\boldsymbol{\phi}_k^*(j))^2\right]$), then the cut-off estimate $\Omega_k(\mathcal{S})$ tends to increase maximally.

While the algorithm in [1] has a goal of finding an $\mathcal{S}$ of smallest possible size that satisfies a target cut-off frequency, we can easily adapt it for our cut-off frequency maximization-based active learning algorithm. This will be discussed in detail in Section 3.1.

2.5 P3: Reconstruction

A graph signal $\mathbf{f} \in PW_\omega(G)$ can be written as a linear combination eigenvectors of $\boldsymbol{\mathcal{L}}$ with eigenvalues less than ω, i.e., $\mathbf{f} = \mathbf{U}_{\mathcal{V},\mathcal{K}}\boldsymbol{\alpha}$ where $\mathcal{K}$ is the index set of those eigenvectors and $\boldsymbol{\alpha}$ is a vector containing the corresponding GFT coefficients. When the unique recovery conditions of Theorem 1 are satisfied, $\boldsymbol{\alpha}$ and the signal $\mathbf{f}$, can be recovered from its subsampled version $\mathbf{f}(\mathcal{S})$ by solving the following least squares problem:

$$\mathbf{f}(\mathcal{S}) = \mathbf{U}_{\mathcal{S},\mathcal{K}}\boldsymbol{\alpha} \tag{11}$$

$$\Rightarrow \boldsymbol{\alpha} = \mathbf{U}_{\mathcal{S},\mathcal{K}}^+ \mathbf{f}(\mathcal{S}). \tag{12}$$

Note that if the original signal $\mathbf{f}$ is not bandlimited, i.e., $\mathbf{f} \notin PW_\omega(G)$, then the least squares solution corresponds to an approximation of $\mathbf{f}$ in $PW_\omega(G)$ (in l_2 sense).

The least squares solution requires eigen-decomposition of $\boldsymbol{\mathcal{L}}$ which is computationally expensive and may not be practical for large graphs. We now describe the iterative, distributed algorithm developed in [18] based on projection onto convex sets (POCS). The proposed method is similar to the Papoulis-Gerchberg algorithm [22] in classical signal processing which is used to reconstruct a bandlimited signal from irregular samples. The convex sets of interest in this case are

$$C_1 = \{\mathbf{x} : \boldsymbol{\mathcal{D}}_\mathcal{S}\mathbf{x} = \boldsymbol{\mathcal{D}}_\mathcal{S}\mathbf{f}\} \tag{13}$$

$$C_2 = PW_\omega(G), \tag{14}$$

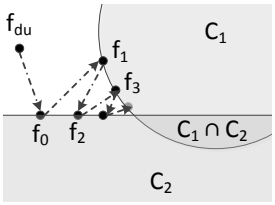


Figure 1: Iterative reconstruction using POCS

where $\boldsymbol{\mathcal{D}}_\mathcal{S}$ is the downsampling operator such that $\boldsymbol{\mathcal{D}}_\mathcal{S}\mathbf{f} = \mathbf{f}(\mathcal{S})$. The unique solution $\mathbf{f}$ to the least squares problem satisfies the following two constraints: (1) the signal equals the known values on the sampling set (i.e., $\mathbf{f} \in C_1$), (2) the signal is ω-bandlimited, where ω is computed using (4) (i.e., $\mathbf{f} \in C_2$). The projector for C_2 is $\boldsymbol{\mathcal{P}}_\omega : \mathbb{R}^N \to PW_\omega(G)$ which is a low-pass graph filter such that

$$\boldsymbol{\mathcal{P}}_\omega \mathbf{x} \in PW_\omega(G) \quad \forall\, \mathbf{x} \in \mathbb{R}^N \tag{15}$$

$\boldsymbol{\mathcal{P}}_\omega$ can be written in graph spectral domain as $\boldsymbol{\mathcal{P}}_\omega = \mathbf{H}(\boldsymbol{\mathcal{L}}) = \sum_{i=1}^N h(\lambda_i)\mathbf{u}_i\mathbf{u}_i^t$ where

$$h(\lambda) = \begin{cases} 1, & \text{if } \lambda < \omega \\ 0, & \text{if } \lambda \geq \omega \end{cases} \tag{16}$$

We define the projection operator for C_1 as $\boldsymbol{\mathcal{P}}_\mathcal{S} : \mathbb{R}^N \to C_1$ which replaces the samples on $\mathcal{S}$ by the known values.

$$\boldsymbol{\mathcal{P}}_\mathcal{S}\mathbf{x} = \mathbf{x} + \boldsymbol{\mathcal{D}}_\mathcal{S}^t(\mathbf{f}(\mathcal{S}) - \boldsymbol{\mathcal{D}}_\mathcal{S}\mathbf{x}). \tag{17}$$

With this notation the proposed iterative algorithm can be written as:

$$\begin{aligned} \mathbf{f}_0 &= \boldsymbol{\mathcal{P}}_\omega(\boldsymbol{\mathcal{D}}_\mathcal{S}^t\mathbf{f}(\mathcal{S})) \\ \mathbf{f}_{i+1} &= \boldsymbol{\mathcal{P}}_\omega\boldsymbol{\mathcal{P}}_\mathcal{S}\mathbf{f}_i \end{aligned} \tag{18}$$

At each iteration the algorithm resets the signal samples on $\mathcal{S}$ to the actual given samples and then projects the signal onto the low-pass space $PW_\omega(G)$. Figure 1 depicts this procedure graphically. It can be shown that $\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{P}}_\omega\boldsymbol{\mathcal{P}}_\mathcal{S}$ is a non-expansive and asymptotically regular operator. Hence, the iterations in (18) converge to the unique point $\mathbf{f} \in C_1 \cap C_2$ which is the desired solution.

The low pass filter $\boldsymbol{\mathcal{P}}_\omega$ above is a spectral graph filter with an ideal brick-wall type spectral response. Thus, the exact computation of $\boldsymbol{\mathcal{P}}_\omega$ would require knowledge of the GFT, which we would like to avoid due to high computational complexity for large graphs. However, it is possible to approximate the ideal filtering operation as a matrix polynomial in terms of $\boldsymbol{\mathcal{L}}$, that can be implemented efficiently using only matrix vector products. Thus we replace $\boldsymbol{\mathcal{P}}_\omega$ in (18) with an approximate low pass filter $\boldsymbol{\mathcal{P}}_\omega^{\text{poly}}$ given by:

$$\boldsymbol{\mathcal{P}}_\omega^{\text{poly}} = \sum_{i=1}^N \left(\sum_{j=0}^p a_j \lambda_i^j \right) \mathbf{u}_i\mathbf{u}_i^t = \sum_{j=0}^p a_j \boldsymbol{\mathcal{L}}^j \tag{19}$$

We specifically use the truncated Chebychev polynomial expansion of any spectral kernel $h(\lambda)$, as proposed in [11], in our experiments. It is easy to show that an operator which is a p-degree polynomial in $\boldsymbol{\mathcal{L}}$ is p-hop localized on the graph and can be implemented in a distributed fashion. In order to ensure that the Chebyshev polynomial approximation is good, we first approximate the ideal spectral kernel by a

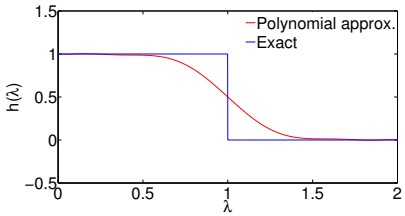


Figure 2: Spectral response of an approximate polynomial filter of degree 10. $\omega = 1, \alpha = 8$.

smooth, continuous sigmoid-like function (see Figure 2)

$$h'(\lambda) = \frac{1}{(1 + \exp(\alpha(\lambda - \omega)))} \tag{20}$$

Due to these approximations in the filter, the reconstructed signal obtained via POCS is different from the true bandlimited signal. However, in semi-supervised learning applications we do not expect the signals (i.e., class membership functions) to be exactly bandlimited anyway. So using a filter with slowly decaying spectral response ends up improving the classification accuracy slightly.

3. GRAPH SAMPLING BASED ACTIVE SEMI-SUPERVISED LEARNING

We now relate the sampling theory developed for graph signals to active semi-supervised learning and propose our solution to the problem. As noted earlier, if the edges of the graph represent similarity between the nodes, then a graph signal defined using the membership functions of a particular class tends to be smooth. This is illustrated experimentally in Figure 3. In Section 2.3 we showed how to estimate the sampling cut-off frequency for a set of vertices. In practice, class membership signals are not strictly bandlimited (see Figure 3). Thus we will be approximating a non-bandlimited signal with one that is bandlimited to the cut-off frequency of the chosen vertex set. The key observation in our work is that, even though we cannot recover the "true" membership signal exactly from its samples, an active learning approach *should aim at selecting the sampling set with maximum cut-off frequency.* This is obviously true since $PW_\omega(G) \subset PW_{\omega'}(G)$ if $\omega \leq \omega'$ and thus, for any signal, its best approximation with a signal from $PW_{\omega'}(G)$ can be no worse (in terms of l_2 error) than its best approximation with a signal from $PW_\omega(G)$.

In this setting, predicting the labels of the unknown datapoints using the labeled data amounts to reconstructing a bandlimited graph signal from its values on the sampling set. Thus, based on the above reasoning the active learning strategy, given a target number of datapoints to be labeled, should be to find a set $\mathcal{S}$, with that size, so that the cut-off frequency of $\mathcal{S}$ is maximized.

3.1 Proposed method

Now, we present the details of our method. We target a multi-class active semi-supervised learning problem with C classes. The true membership function for class j is denoted as $\mathbf{f}_j : \mathcal{V} \mapsto \{0, 1\}$, where $\mathbf{f}_j(i) = 1$ indicates that node i belongs to class j. These membership functions are taken to be the graph signals for our setting. The predicted membership functions for each class take real values and are denoted as $\hat{\mathbf{f}}_j : \mathcal{V} \mapsto \mathbb{R}$. The predicted label of node i is given by $\arg\max_j \hat{\mathbf{f}}_j(i)$. We denote the labeled set as $\mathcal{S}_L$ and the unlabeled set as $\mathcal{S}_U = \mathcal{V} \setminus \mathcal{S}_L$. Then, our solution to the active semi-supervised learning task can be formally summarized as follows:

Algorithm 1 Greedy heuristic for finding $\mathcal{S}_L^*$

Input: $G = \{\mathcal{V}, E\}$, $\boldsymbol{\mathcal{L}}$, target size m, parameter $k \in \mathbb{Z}^+$.
Initialize: $\mathcal{S} = \{\emptyset\}$.
1: **while** $|\mathcal{S}| \leq m$ **do**
2: For $\mathcal{S}$, compute the smoothest signal $\phi_k^* \in L_2(\mathcal{S}^c)$ using (4) and (5).
3: $v \leftarrow \arg\max_i \left[(\phi_k^*(i))^2\right]$.
4: $\mathcal{S} \leftarrow \mathcal{S} \cup v$.
5: **end while**
6: $\mathcal{S}_L^* \leftarrow \mathcal{S}$.

1. Given a size m and parameter k, we first find the optimal labeled set $\mathcal{S}_L^*$ and corresponding cut-off frequency $\Omega_k(\mathcal{S}_L^*)$ as follows:

$$\mathcal{S}_L^* = \underset{\mathcal{S}:|\mathcal{S}|=m}{\arg\max}\ \Omega_k(\mathcal{S}) \tag{21}$$

   We solve this problem in a greedy fashion by adding nodes to $\mathcal{S}$ that maximize the increase in $\Omega_k(\mathcal{S})$ at each step (cf. Section 2.4). This procedure is summarized with Algorithm 1
2. Next, we query the labels of nodes in $\mathcal{S}_L^*$.
3. Finally, we determine the predicted membership functions $\hat{\mathbf{f}}_j$ for each class from $\mathbf{f}_j(\mathcal{S}_L^*), j = 1, \ldots, C$ using the POCS iterative method described in Section 2.5, where $\mathcal{S} = \mathcal{S}_L^*$ and $\omega = \Omega_k(\mathcal{S}_L^*)$ are used in (13) and (14) to construct the convex sets.

3.2 Graph Theoretic Interpretation

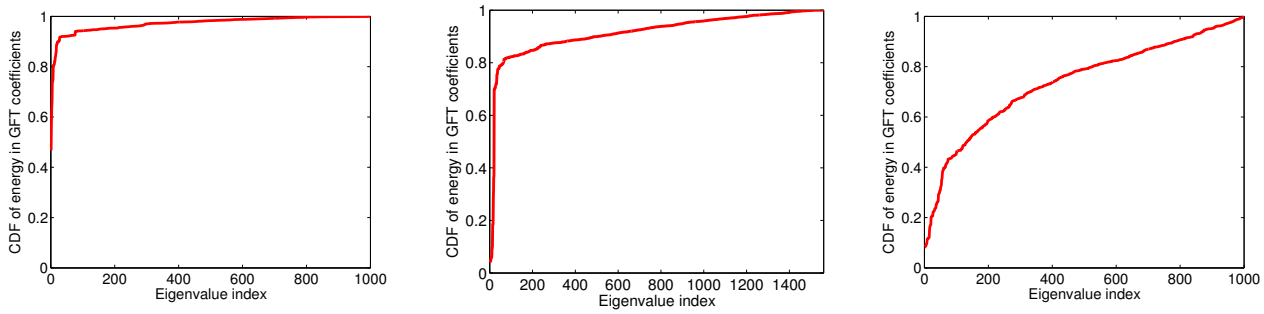
In this section, we will provide an intuitive interpretation for our node selection algorithm in terms of connected-ness among the nodes. To simplify the exposition, we consider the maximization problem (21) for $k = 1$:

$$\Omega_1(\mathcal{S}) = \inf_{\substack{\mathbf{x}(\mathcal{S})=\mathbf{0} \\ ||\mathbf{x}||=1}} \mathbf{x}^t \boldsymbol{\mathcal{L}} \mathbf{x} \tag{22}$$

This expression appears more commonly as part of discrete Dirichlet eigenvalue problems on graphs. Specifically, it is equal to the Dirichlet energy of the subset $\mathcal{S}^c$ [6, 19]. The sampling set selection problem seeks to identify the subset $\mathcal{S}$ that maximizes this objective function. To give an intuitive interpretation of our goal, we expand the objective function for any $\mathbf{x}$ with constraint $\mathbf{x}(\mathcal{S}) = \mathbf{0}$ as follows:

$$\begin{aligned}\mathbf{x}^t \boldsymbol{\mathcal{L}} \mathbf{x} &= \sum_{i \sim j} w_{ij} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2 \\ &= \sum_{\substack{i \sim j \\ i \in \mathcal{S}, j \in \mathcal{S}^c}} w_{ij} \left(\frac{x_j^2}{d_j} \right) + \sum_{\substack{i \sim j \\ i,j \in \mathcal{S}^c}} w_{ij} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2 .\end{aligned} \tag{23}$$

The minimizer in the equation above is the first Dirichlet eigenvector which is guaranteed to have strictly positive values on $\mathcal{S}^c$ [19]. Therefore, the contribution of the second



(a) USPS (b) Isolet (c) 20 newsgroups

Figure 3: Cumulative distribution of energy in the GFT coefficients of one of the class membership functions pertaining to the three real-world dataset experiments considered in Section 5. Note that most of the energy is concentrated in the low-pass region.

term is expected to be negligible as compared to the first one due to differencing, and we get

$$\mathbf{x}^t \boldsymbol{\mathcal{L}} \mathbf{x} \approx \sum_{j \in \mathcal{S}^c} \left(\frac{p_j}{d_j} \right) x_j^2, \tag{24}$$

where, $p_j = \sum_{i \in \mathcal{S}} w_{ij}$ is defined as the "partial out-degree" of node $j \in \mathcal{S}^c$, i.e., it is the sum of weights of edges crossing over to the set $\mathcal{S}$. Therefore, given a current selected $\mathcal{S}$, the greedy algorithm selects the next node, to be added to $\mathcal{S}$, that maximizes the increase in

$$\Omega_1(\mathcal{S}) \approx \inf_{||\mathbf{x}||=1} \sum_{j \in \mathcal{S}^c} \left(\frac{p_j}{d_j} \right) x_j^2. \tag{25}$$

Due to the constraint $||\mathbf{x}|| = 1$, the expression being minimized is essentially an infimum over a convex combination of the fractional out-degrees and its value is largely determined by nodes $j \in \mathcal{S}^c$ for which p_j/d_j is small. In other words, we must worry about those nodes that have a low ratio of partial degree to the actual degree. Thus, in the simplest case, our selection algorithm tries to remove those nodes from the unlabeled set that are weakly connected to nodes in the labeled set. This makes intuitive sense as, in the end, most prediction algorithms involve propagation of labels from the labeled to the unlabeled nodes. If an unlabeled node is strongly connected to various numerous points, its label can be assigned with greater confidence.

Note that using a higher power k in the cost function, i.e., finding $\Omega_k(\mathcal{S})$ for $k > 1$ involves $\mathbf{x}\boldsymbol{\mathcal{L}}^k\mathbf{x}$ which, loosely speaking, takes into account higher order interactions between the nodes while choosing the nodes to label. In a sense, we expect it to capture the connectivities in a more *global* sense, beyond local interactions, taking into account the underlying manifold structure of the data.

3.3 Complexity

We now comment on the time and space complexity of our algorithm. The most complex step in the greedy procedure for maximizing $\Omega_k(\mathcal{S})$ is computing the smallest eigenpair of $(\boldsymbol{\mathcal{L}}^k)_{\mathcal{S}^c}$. This can be accomplished using an iterative Rayleigh-quotient minimization based algorithm. Specifically, the locally-optimal pre-conditioned conjugate gradient (LOPCG) method [14] is suitable for this approach. Note that $(\boldsymbol{\mathcal{L}}^k)_{\mathcal{S}^c}$ can be written as $\mathbf{I}_{\mathcal{S}^c,\mathcal{V}}.\boldsymbol{\mathcal{L}}.\boldsymbol{\mathcal{L}} \ldots \boldsymbol{\mathcal{L}}.\mathbf{I}_{\mathcal{V},\mathcal{S}^c}$, hence the eigenvalue computation can be broken into atomic matrix-vector products: $\boldsymbol{\mathcal{L}}.\mathbf{x}$. Typically, the graphs encountered in learning applications are sparse, leading to efficient implementations of $\boldsymbol{\mathcal{L}}.\mathbf{x}$. If $|\boldsymbol{\mathcal{L}}|$ denotes the number of non-zero elements in $\boldsymbol{\mathcal{L}}$, then the complexity of the matrix-vector product is $O(|\boldsymbol{\mathcal{L}}|)$. The complexity of each eigen-pair computation for $(\boldsymbol{\mathcal{L}}^k)_{\mathcal{S}^c}$ is then $O(k|\boldsymbol{\mathcal{L}}|r)$, where r is a constant equal to the average number of iterations required for the LOPCG algorithm (r depends on the spectral properties of $\boldsymbol{\mathcal{L}}$ and is independent of its size $|\mathcal{V}|$). The complexity of the label selection algorithm then becomes $O(k|\boldsymbol{\mathcal{L}}|mr)$, where m is the number of labels requested.

In the iterative reconstruction algorithm, since we use polynomial graph filters (Section 2.5), once again the atomic step is the matrix-vector product $\boldsymbol{\mathcal{L}}.\mathbf{x}$. The complexity of this algorithm can be given as $O(|\boldsymbol{\mathcal{L}}|pq)$, where p is the order of the polynomial used to design the filter and q is the average number of iterations required for convergence. Again, both these parameters are independent of $|\mathcal{V}|$. Thus, the overall complexity of our algorithm is $O(|\boldsymbol{\mathcal{L}}|(kmr + pq))$. In addition, our algorithm has major advantages in terms of space complexity: Since, the atomic operation at each step is the matrix-vector product $\boldsymbol{\mathcal{L}}.\mathbf{x}$, we only need to store $\boldsymbol{\mathcal{L}}$ and a constant number of vectors. Moreover, the structure of the Laplacian matrix allows one to perform the aforementioned operations in a distributed fashion. This makes it well-suited for large-scale implementations using software packages such as GraphLab [16].

3.4 Prediction Error and Number of Labels

As discussed in Section 2.5, given the samples $\mathbf{f}_{\mathcal{S}}$ of the true graph signal on a subset of nodes $\mathcal{S} \subset \mathcal{V}$, its estimate on $\mathcal{S}^c$ is obtained by solving the following problem:

$$\hat{\mathbf{f}}(\mathcal{S}^c) = \mathbf{U}_{\mathcal{S}^c,\mathcal{K}}\boldsymbol{\alpha}^* \text{ where, } \boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{U}_{\mathcal{S},\mathcal{K}}\boldsymbol{\alpha} - \mathbf{f}(\mathcal{S})\| \tag{26}$$

Here, $\mathcal{K}$ is the index set of eigenvectors with eigenvalues less than the cut-off $\omega_c(\mathcal{S})$. If the true signal $\mathbf{f} \in PW_{\omega_c(\mathcal{S})}(G)$, then the prediction is perfect. However, this is not the case in most problems. The prediction error $\|\mathbf{f} - \hat{\mathbf{f}}\|$ roughly equals the portion of energy of the true signal in $[\omega_c(\mathcal{S}), \lambda_N]$ frequency band. By choosing the sampling set $\mathcal{S}$ that maximizes $\omega_c(\mathcal{S})$, we try to capture most of the signal energy and thus, reduce the prediction error.

An important question in the context of active learning is determining the minimum number of labels required so that

the prediction error $\|\mathbf{f}-\hat{\mathbf{f}}\|$ is less that some given tolerance δ. To find this we first characterize the smoothness $\gamma(\mathbf{f})$ of a signal $\mathbf{f}$ as

$$\gamma(\mathbf{f}) = \min \theta \;\text{ s.t. }\; \|\mathbf{f} - \boldsymbol{\mathcal{P}}_\theta \mathbf{f}\| \leq \delta$$

The following theorem gives a lower bound on the minimum of number of labels required in terms of $\gamma(\mathbf{f})$.

THEOREM 2. *If $\hat{\mathbf{f}}$ is obtained by solving* (26)*, then the minimum number of labels l required to satisfy $\|\mathbf{f}-\hat{\mathbf{f}}\| \leq \delta$ is greater than p, where p is the number of eigenvalues of $\boldsymbol{\mathcal{L}}$ less than $\gamma(\mathbf{f})$.*

PROOF. In order for (26) to have a unique solution, $\mathbf{U}_{\mathcal{S},\mathcal{K}}$ needs to have full column rank, which implies that $l = |\mathcal{S}| \geq |\mathcal{K}|$. Now, for $\|\mathbf{f}-\hat{\mathbf{f}}\| \leq \delta$ to hold the bandwidth of $\hat{\mathbf{f}}$ has to be at least $\gamma(\mathbf{f})$, or in other words, $|\mathcal{K}| \geq p$. This gives us the desired result as $l \geq |\mathcal{K}| \geq p$. □

4. RELATED WORK

Different frameworks have been proposed for pool-based batch-mode active semi-supervised learning including optimal experiment design [27, 26], generalization error bound minimization [7, 8] and submodular optimization [9, 10, 12]. We now point out connections between some of the graph based approaches in the above categories and our graph signal sampling theory based framework.

The notion of frequency given by GFT is closely related to Laplacian eigenmaps which is a well known dimensionality reduction technique [2]. GFT can be viewed as a way of measuring the signal variation on the manifold represented by Laplacian eigenmaps. By selecting nodes that maximize the bandwidth of the space of recoverable signals, we are trying to capture as many dimensions of the manifold structure of the data with as few samples as possible. A related active learning method proposed by Zhang et al. [27] uses optimal experiment design while considering local structure of the data in a way which is similar to local linear embedding (LLE) for approximating the underlying low-dimensional manifold [21]. This approach tries to choose the most representative data points from which one can recover the whole data set by local linear reconstruction. It is interesting to note that under certain conditions LLE and Laplacian eigenmaps are equivalent [15].

Gu and Han [7] propose a method based on minimizing the generalization error bound for learning with local and global consistency (LLGC) [28]. Their formulation boils down to choosing subset S that minimizes $\mathrm{Tr}\left((\mu\mathbf{L}_S + \mathbf{I})^{-2}\right)$. To relate this formulation to our proposed method, note that

$$\mathrm{Tr}\left((\mu\mathbf{L}_S + \mathbf{I})^{-2}\right) = \sum_i \frac{1}{(\zeta_i+1)^2} \leq \frac{|S|}{(\zeta_1+1)^2}$$

where, $\zeta_1 \leq \ldots \leq \zeta_{|S|}$ denote the eigenvalues of $\mathbf{L}_S$. Loosely speaking, minimizing the above objective function is equivalent to maximizing the smallest eigenvalue ζ_1 of $\mathbf{L}_S$. So, this method essentially tries to ensure that the labeled set is well-connected to the unlabeled set whereas our method ensures that the unlabeled set is well-connected to the labeled set (cf. Section 3.2).

Submodular functions have been used for active semi-supervised learning on graphs by Guillory and Bilmes [10, 9]. In this work, the subset of nodes $S \subset \mathcal{V}$ is chosen to maximize

$$\Psi(S) = \min_{T \subseteq \mathcal{V}\setminus S : T \neq \emptyset} \frac{\Gamma(T)}{|T|}, \tag{27}$$

where $\Gamma(T)$ denotes the cut function $\sum_{i\in T, j\notin T} w_{ij}$. Intuitively, maximizing $\Psi(S)$ ensures that no subset of unlabeled nodes is weakly connected to the labeled set S. This agrees with the graph theoretic interpretation of our method given in Section 3.2. They also provide a bound on the prediction error in terms $\Psi(S)$ and a smoothness function $\Phi(\mathbf{f}) = \sum_{i,j} w_{ij}|f_i - f_j|$. This bound gives a theoretical justification for semi-supervised learning using min-cuts [4]. It also motivates a graph partitioning-based active learning heuristic [9] which says that to select l nodes to label, the graph should be partitioned into l clusters and one node should be picked at random from each cluster.

5. EXPERIMENTS

We compare our method against three active semi-supervised learning approaches mentioned in the previous section, namely, LLR [27], LLGC error bound minimization [7], METIS graph partitioning based heuristic [9] and Ψ-max [10]. The details of implementation of each method are as follows:

1. The LLR approach [27] allows any prediction method once the samples to be queried are chosen. We use the Laplacian regularized least squares (LapRLS) [3] method for prediction (used in [27]).

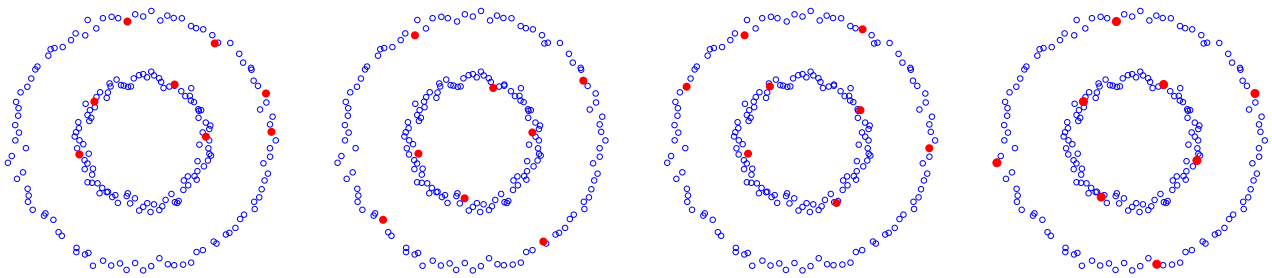
2. In our implementation of the LLGC bound method [7], we fix the parameter μ to 0.01. Since this approach is based on minimizing the generalization error bound for LLGC, we use the same method for prediction with the queried samples. [1]

3. The normalized cut based active learning heuristic of Guillory and Bilmes [9] is implemented using the METIS graph partitioning package [13]. This algorithm chooses a random node to label from each partition, so we average the error rates over a 100 trials.

In the implementation of our proposed method, we use approximate polynomial filters of degree 10 with $\alpha = 8$. The parameter k in our method is fixed as 8 for these experiments. Its effect on classification accuracy is studied in Section 5.4. In addition to the above methods, we also compare with the random sampling strategy. We use LapRLS to predict the unknown labels from the randomly queried samples and report the average error rates over 30 trials.

To intuitively demonstrate the effectiveness of our method, we first test it on a two circles toy data as shown in Figure 4. The data is comprised of 200 nodes from which we would like to select 8 nodes to query. We construct a weighted sparse graph by connecting each node to its 10 nearest neighbors while ensuring that the connections are symmetric. The edge weights are computed with the Gaussian kernel $\exp\left(-\frac{||\mathbf{x}_i-\mathbf{x}_j||^2}{2\sigma^2}\right)$ (except in the case of Ψ-max where the graph is unweighted). It can be seen from Figure 4 that all the methods choose 4 points from each of the two circles. Additionally, the proposed approach selects evenly

[1] In our experiments, we observed that the greedy algorithm given in [7] did not converge to a good solution. So we use Monte-Carlo simulations to minimize the objective function.



(a) Ψ-max (b) LLR (c) LLGC Bound (d) Proposed

Figure 4: Toy example comparing the nodes selected using different active learning methods

spaced data points within one circle, while at the same time maximizing the spacing between the selected data points in different circles. This is in accordance with the requirement of choosing points which are most representative of the data.

We tested our method in three application scenarios: Handwritten digit recognition, text classification and spoken letters recognition. In these experiments, we do not compare with Ψ-max since the method has computational complexity of $O(N^6)$ and, to the best of our knowledge, is not scalable. Next, we provide the details of each experiment. Both the datasets and the graph construction procedures used are typical of what has been used in the literature.

5.1 Handwritten digits classification

In this experiment, we used our proposed active semi-supervised learning algorithm to perform a classification task on the USPS handwritten digits dataset[2]. This dataset consists of 1100 16×16 pixel images for each of the digits 0 to 9. We used 100 randomly selected samples for each digit class to create one instance of our dataset. Thus each instance consists of 1000 feature vectors (100 samples/class $\times$ 10 digit classes) of dimension 256.

The graph is constructed using Gaussian kernel weights $w_{ij} = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right)$, where $\mathbf{x}_i$ is the 256-dimensional feature vector composed of pixel intensity values for each image. The parameter σ is chosen to be 1/3-rd of the average distance to the K-th nearest neighbor for all datapoints. This heuristic has been suggested in [5]. We fix $K = 10$. Additionally, the graph is sparsified approximately by restricting the connectivity of each datapoint to its K nearest neighbors, i.e., an edge between nodes i and j is removed unless node i is among the K-nearest neighbors of node j or vice-versa. This results in a symmetric adjacency matrix for the graph. Using the graph constructed, we select the points to label and report prediction error after reconstruction using our semi-supervised learning algorithm. We repeat the classification over 10 such instances of the dataset and report the average classification error. The results are illustrated in Figure (5a). We observe that our proposed method outperforms the others. A notable feature of our method is that we show very good classification results even for very few labeled samples. This is due to our inherent criterion for active learning that tries to select those points that maximize the recoverable dimensions of the underlying data manifold.

[2] `http://www.cs.nyu.edu/~roweis/data.html`

5.2 Text classification

For our text classification example, we use the 20 newsgroups dataset[3]. It contains around 20,000 documents, partitioned in 20 different newsgroups. For our experiment, we consider 10 groups of documents, namely, {comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, rec.autos, rec.motorcycles, sci.crypt, sci.electronics, sci.med, sci.space}, and randomly choose 100 datapoints from each group. We generate 10 such instances of 1000 data points each and report the average errors. We clean the dataset by removing the words that appear in fewer than 20 documents and then select only the 3000 most frequent ones from the remaining words. To form the feature vectors representing the documents, we use the tf-idf statistic of these words. The tf-idf statistic captures the relative importance of a word in a document in a corpus:

$$\text{tf-idf} = (1 + \log(\text{tf})) \times \log\left(\frac{N}{\text{idf}}\right) \tag{28}$$

where, tf is the frequency of a word in a document, idf is the number of documents in which the word appears and N is the total number of documents. Thus, we get 1000 feature vectors in 3000 dimensional space. To form the graph of documents, we compute the pairwise cosine similarity between their feature vectors. Each node is connected to the 10 nodes that are most similar to it and the resultant graph is then symmetrized. The classification results in Figure (5b) show that our method performs very well compared to others. However, the absolute error rates are not very good. This is due to the high similarity between different newsgroups which makes the problem inherently difficult.

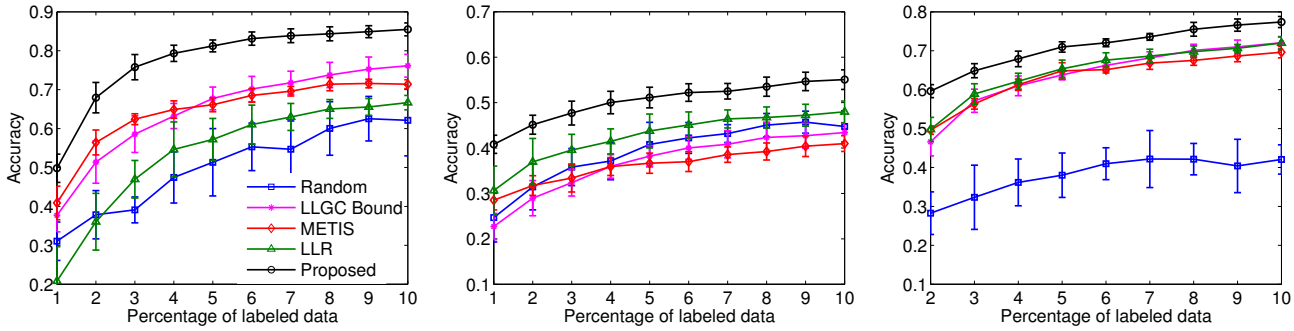
5.3 Spoken letters classification

For the spoken letters classification example, we considered the Isolet dataset[4]. It consists of letters of the English alphabet spoken in isolation twice by 150 different subjects. The speakers are grouped into 5 sets of 30 speakers each, with the groups referred to as isolet1 through isolet5. Each alphabet utterance has been pre-processed beforehand to create a 617-dimensional feature vector.

For this experiment, we considered the task of active semi-supervised classification of utterances into the 26 alphabet categories. To form an instance of the dataset, 60 utterances are randomly selected out of 300 for each alphabet. Thus, each instance consists of $60 \times 26 = 1560$ datapoints

[3] `http://qwone.com/~jason/20Newsgroups/`

[4] `http://archive.ics.uci.edu/ml/datasets/ISOLET`



(a) USPS (b) 20 newsgroups (c) Isolet

Figure 5: Comparison of active semi-supervised learning methods on real datasets. Plots show the average classification accuracies for different percentages of labeled data.

of dimension 617. As in the hand-written digits classification problem, the graph is constructed using Gaussian kernel weights between nodes, with σ taken as 1/3-rd of the average distance to the K-th nearest neighbor for each datapoint. We select $K = 10$ for our experiment. Sparsification of the graph is carried out approximately using K-nearest neighbor criterion. With the constructed graph, we perform active semi-supervised learning using all the methods. The experiment is repeated over 10 instances of the dataset and average prediction error is reported in Figure (5c). Note that we start with 2% labeled points to ensure that each method gets a fair chance of selecting at least one point to label from each of the 26 classes. We observe that our method outperforms the others.

5.4 Effect of parameter k

To study the effect of parameter k in the proposed method on classification accuracy we repeat the above experiments for different values of k. Figure 6 shows the results. For the USPS and Isolet datasets, the classification accuracies remain largely unchanged for different values of k. For the 20 Newsgroups dataset, a slight improvement in classification accuracies is observed for higher values of k. This result agrees with the distribution of GFT coefficients of the class membership functions in each dataset shown in Figure 3. In USPS and Isolet datasets, most of the energy of the graph signal (i.e., the class membership functions) is contained in the first few frequencies. Thus, increasing the value of k, so that a better estimate of cut-off frequency is maximized during the choice of sampling set, is not necessary. In other words, maximizing a loose estimate of the cut-off frequency is sufficient. However, the membership functions in the 20 Newsgroups dataset have a significant fraction of their energy spread over high frequencies as shown in Figure 3. Due to this, maximizing a tighter estimate of the the cut-off allows the sampling set selection algorithm to pick nodes that capture more signal energy, resulting in higher accuracies.

6. CONCLUSION

In this paper, we introduce a novel framework for batch mode active semi-supervised learning based on sampling theory for graph signals. The proposed active learning framework aims to select the subset nodes which maximizes the dimension of the space of uniquely recoverable signals. In the context of sampling theory, this translates to selecting the subset with the maximum cut-off frequency. This interpretation leads to a very efficient greedy algorithm. We provide intuition about how the method tries to choose the nodes which are most representative of the data. We also present an efficient semi-supervised learning method based on bandlimited interpolation. We show, through experiments on real data, that our two algorithms, in conjunction, perform very well compared to state of the art methods.

In the future, we would like to provide bounds on the prediction error of the proposed method (when the true signal is not exactly bandlimited) in terms of signal smoothness and the cut-off frequency. We also hope to have tighter bounds on the number of labels required for desired prediction accuracy. It would be useful to consider an extension of the proposed framework to a partially batch setting so that we can incorporate the label information from previous batches to improve the choice of sampling sets.

7. REFERENCES

[1] A. Anis, A. Gadde, and A. Ortega. Towards a sampling theorem for signals on arbitrary graphs. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014*.

[2] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine learning*, 56(1-3):209–239, 2004.

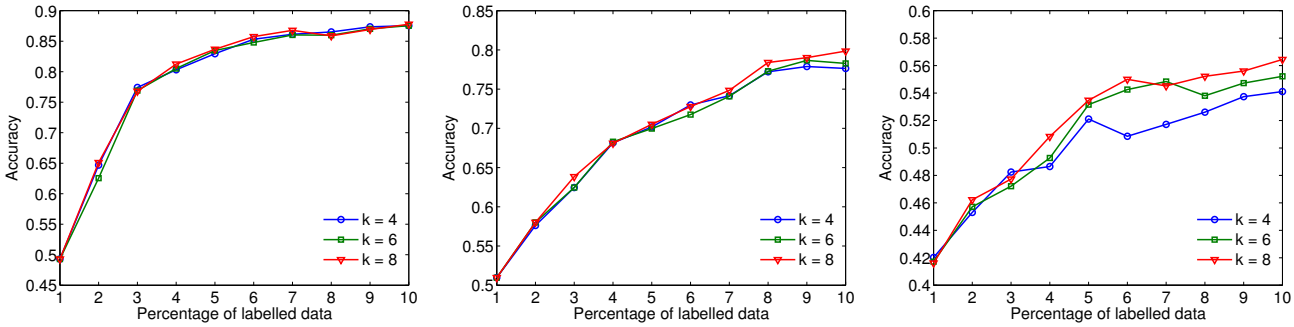
[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006.

[4] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning*, pages 19–26, 2001.

[5] O. Chapelle, B. Schölkopf, A. Zien, et al. *Semi-supervised learning*. MIT Press, Cambridge, 2006.

[6] F. R. K. Chung. *Spectral graph theory*, volume 92. CBMS Regional Conference Series in Mathematics, AMS, 1997.

[7] Q. Gu and J. Han. Towards active learning on graphs: An error bound minimization approach. In *Proceedings of 12th IEEE International Conference on Data Mining*, pages 882–887, 2012.



(a) USPS (b) Isolet (c) 20 newsgroups

Figure 6: Effect of k on classification accuracy of the proposed method. Plots show the average classification accuracy for different percentages of labelled data.

[8] Q. Gu, T. Zhang, C. Ding, and J. Han. Selective labeling via error bound minimization. In *Advances in Neural Information Processing Systems 25*, pages 332–340. 2012.

[9] A. Guillory and J. Bilmes. Label selection on graphs. In *Advances in Neural Information Processing Systems 22*, pages 691–699. 2009.

[10] A. Guillory and J. Bilmes. Active semi-supervised learning using submodular functions. In *Proceedings of 27th Conference on Uncertainty in Artificial Intelligence*, pages 274–282, 2011.

[11] D. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129 – 150, 2011.

[12] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International conference on Machine learning*, pages 417–424, 2006.

[13] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1), 1998.

[14] A. V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM journal on scientific computing*, 23(2):517–541, 2001.

[15] D. Kong, C. H. Ding, H. Huang, and F. Nie. An iterative locally linear embedding algorithm. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[16] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein. Distributed graphlab: A framework for machine learning and data mining in the cloud. *Proc. VLDB Endow.*, 5(8):716–727, Apr. 2012.

[17] S. Narang and A. Ortega. Perfect reconstruction two-channel wavelet filter banks for graph structured data. *IEEE Transactions on Signal Processing*, 60(6):2786–2799, June 2012.

[18] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega. Localized iterative methods for interpolation in graph structured data. In *Signal and Information Processing (GlobalSIP), 2013 IEEE Global Conference on*, 2013.

[19] B. Osting, C. D. White, and E. Oudet. Minimal Dirichlet energy partitions for graphs. Aug. 2013. arXiv:1308.4915 [math.OC].

[20] I. Pesenson. Sampling in Paley-Wiener spaces on combinatorial graphs. *Transactions of the American Mathematical Society*, 360(10):5603–5627, 2008.

[21] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[22] K. Sauer and J. Allebach. Iterative reconstruction of bandlimited images from nonuniformly spaced samples. *Circuits and Systems, IEEE Transactions on*, 34(12):1497–1506, 1987.

[23] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2010.

[24] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. Signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular data domains. *Signal Processing Magazine, arXiv:1211.0053*, May. 2013.

[25] A. J. Smola and R. Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.

[26] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd International conference on Machine learning*, pages 1081–1088, 2006.

[27] L. Zhang, C. Chen, J. Bu, D. Cai, X. He, and T. S. Huang. Active learning based on locally linear reconstruction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(10):2026–2038, 2011.

[28] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*. 2004.

[29] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2008.

[30] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.