Online Graph Spectra Learning

Stefan Dernbach and Don Towsley
College of Information and Computer Sciences
University of Massachusetts
Amherst, MA, USA 01003
{dernbach,towsley}@cs.umass.edu

***Abstract*— Spectral graph signal processing techniques use the eigenvectors of the graph Laplacian to transform a graph signal from the spatial domain to the frequency domain of the graph. In general eigendecompositons requires knowledge of the entire graph structure to compute. In cases where this isn't possible, spectral signal processing techniques can't be readily applied. We propose an iterative method of calculating the spectra of the graph Laplacian from sampling edges of the graph.**

I. INTRODUCTION

The graph Fourier transform is a pivotal tool in spectral graph signal processing techniques [1] as well as spectral graph convolution neural networks [2]. These techniques utilize the spectral decomposition of the graph Laplacian (or similar matrix) to define this transform. This decomposition requires knowledge of the entire graph to perform; a requirement not always practical when dealing with excessively large, dynamic, or obfuscated graphs. In this work we propose an iterative, online method to learn the eigenvectors of the graph Laplacian from single edge samples of the graph. Our technique does not require storing any information about the graph aside from its size, and current eigenvector estimates.

The following notation and definitions are used throughout the paper. Given an undirected graph $G{=}(V, E)$ composed of a set of N vertices V, and M edges E, the matrices $\mathbf{A}$ and $\mathbf{D}$ denote the adjacency and diagonal degree matrices respectively. The graph Fourier transform transforms a signal $\mathbf{x} \in \mathbb{R}^N$ from the spatial domain into the frequency domain of the graph: $\hat{f}_G(\mathbf{x}) = \mathbf{U}^T\mathbf{x}$ where $\mathbf{U}$ is the matrix whose columns are composed of the eigenvectors of the graph Laplacian, $\mathbf{L} = \mathbf{D} - \mathbf{A}$. We order the eigenvectors, $\mathbf{u}_i$, according to their corresponding eigenvalues, λ_i, from smallest eigenvalue to largest. We assume G is connected so that $0 = \lambda_0 < \lambda_1$ and $\mathbf{u}_0$ aligns with the vector of all ones. Of interest are eigenvectors $\{\mathbf{u}_i : i = 1, ..., N{-}1\}$.

II. METHOD

Given a graph G, the incidence matrix $\mathbf{B} \in \mathbb{R}^{M\times N}$ relates edges to their incident vertices by:

$$\mathbf{B}_k = \mathbf{e}_i - \mathbf{e}_j \tag{1}$$

where $\mathbf{e}_i$ is a vector of zeros with a 1 in the i-th position and (v_i, v_j) is the k-th edge in E. It can be easily verified that $\mathbf{L} = \mathbf{B}^T\mathbf{B}$. Equivalently, this can be rewritten as a sum of the outer products of the rows of $\mathbf{B}$:

$$\mathbf{L} = \sum_{(v_i,v_j)\in E} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T. \tag{2}$$

It follows from (2) that for a randomly sampled edge (v_i, v_j) the outer product:

$$\tilde{\mathbf{L}}_{ij} = (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T \tag{3}$$

is an unbiased estimate of $\mathbf{L}/M$; the mean of $\tilde{\mathbf{L}}_{ij}$ over all $(v_i, v_j) \in E$ is exactly $\mathbf{L}/M$. This estimate forms the basis for our online learning approach because $\mathbf{L}$ and $\mathbf{L}/M$ have the same eigenvectors with scaled eigenvalues. Note that the ordering of the nodes in an edge does not affect the outer product: $(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T = (\mathbf{e}_j - \mathbf{e}_i)(\mathbf{e}_j - \mathbf{e}_i)^T$.

In the setting where the entire graph is known, the Rayleigh quotient

$$r = \frac{\mathbf{w}^T\mathbf{L}\mathbf{w}}{\mathbf{w}^T\mathbf{w}}, \tag{4}$$

with $\mathbf{w}$ denoting the current eigenvector approximation, can be used as a loss function in gradient descent to learn the eigenvectors of $\mathbf{L}$ as an alternative to performing a singular value decomposition. In the online setting, we replace $\mathbf{L}$ with each estimate $\tilde{\mathbf{L}}_{ij}$ as we sample edges (v_i, v_j). This provides an update equation for $\mathbf{w}$:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \alpha\left(\tilde{\mathbf{L}}_{ij}\mathbf{w}^{(t)} - (\mathbf{w}^{(t)})^T\tilde{\mathbf{L}}_{ij}\mathbf{w}^{(t)}\mathbf{w}^{(t)}\right) \tag{5}$$

that is similar to the Oja update rule for minor component analysis [3].

The sparse nature of $\tilde{\mathbf{L}}_{ij}$ allow us to reduce many of the matrix and vector products to scalar arithmetic leading to Algorithm 1 below. Multiple eigenvectors can be learned simultaneously by including orthogonalization steps (i.e. Grahm-Schmidt) intermittently during the process. The trivial eigenvector $\mathbf{u}_0$ can be ignored in the learning by mean centering $\mathbf{w}$ as an efficient method of orthogonalizing the vector to $\mathbf{u}_0$.

The update in (5) can be trivially repurposed to learn eigenvectors of the symmetric normalized graph Laplacian, $\hat{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$, if the degree of the nodes incident to the sampled edge are given. Changing (3) to:

$$\tilde{\mathbf{L}}_{ij} = \left(\mathbf{D}_{ii}^{-\frac{1}{2}}\mathbf{e}_i - \mathbf{D}_{jj}^{-\frac{1}{2}}\mathbf{e}_j\right)\left(\mathbf{D}_{ii}^{-\frac{1}{2}}\mathbf{e}_i - \mathbf{D}_{jj}^{-\frac{1}{2}}\mathbf{e}_j\right)^T \tag{6}$$

gives an unbiased estimate of $\hat{\mathbf{L}}/M$.

Algorithm 1 Online Graph Spectra

Input: α, N
Initialize $\mathbf{w}^{(0)} \in \mathbb{R}^N$
for $n = 0$ **to** N **do**
  Sample edge (v_i, v_j) from $\mathbf{G}$
  $y^{(n)} \leftarrow \mathbf{w}_i^{(n)} - \mathbf{w}_j^{(n)}$
  $\Delta\mathbf{w}^{(n)} \leftarrow -(y^{(n)})^2 * \mathbf{w}^{(n)} + y^{(n)}\mathbf{e}_i - y^{(n)}\mathbf{e}_j$
  $\mathbf{w}^{(n+1)} \leftarrow \mathbf{w}^{(n)} - \alpha\Delta\mathbf{w}^{(n)}$
end for

III. EVALUATION

We evaluate our method using two error metrics that highlight different forms of error in the approximated eigenvectors. Given an eigenvector estimate $\mathbf{w}$ (normalized to unit length), and actual eigenvalue λ the normalized eigenvalue error is evaluated as

$$\text{error}(\mathbf{w}, \lambda) = \frac{|\mathbf{w}^T\mathbf{L}\mathbf{w} - \lambda|}{\lambda}. \tag{7}$$

The second evaluation metric is the cosine distance between the estimated eigenvector and the actual eigenvector $\mathbf{u}$,

$$\text{dist}(\mathbf{w}, \mathbf{u}) = 1 - \mathbf{w}^T\mathbf{u}, \tag{8}$$

These metrics emphasize different errors in the estimates. The value $\mathbf{w}^T\mathbf{L}\mathbf{w}$ in (7) can be written in point wise form as $\sum_{(v_i,v_j)\in E} (\mathbf{w}_i - \mathbf{w}_j)^2$ which highlights that errors in high degree nodes have a larger effect than low degree nodes because they appear more often in the summation. Conversely, a node in the graph with only a single edge can have a very large error in the eigenvector without significantly affecting the eigenvalue. The cosine distance measure, on the other hand, weights all nodes in the graph equally for purposes of evaluating the eigenvector estimate. In effect, the eigenvalue difference is an edge focused error metric, while the cosine distance is a node focused one.

We evaluate our method on learning the first five eigenvectors of the Zachary Karate Club Network [4] composed of 34 nodes and 78 edges. Edges are sampled uniformly with replacement. Figure 1 shows the convergence of our method as we sample edges from the graph. We compare against learning the eigenvectors using gradient descent on the entire graph Laplacian. We use the Grahm-Schmidt process to orthogonalize the vectors in both methods. Orthogonalization is performed an equal number of times for both methods by processing the vectors after every batch update and after every 78 edge samples for the online method. Reported errors are averaged over five trials.

Errors are comparable between both the online method and the batch setting. Noticeably, the online method appears slightly better by the eigenvalue metric and slightly worse in the cosine metric for lower eigenvectors. This is unsurprising given that edges are uniformly sampled leading, leading to a larger number of updates of the high degree nodes.

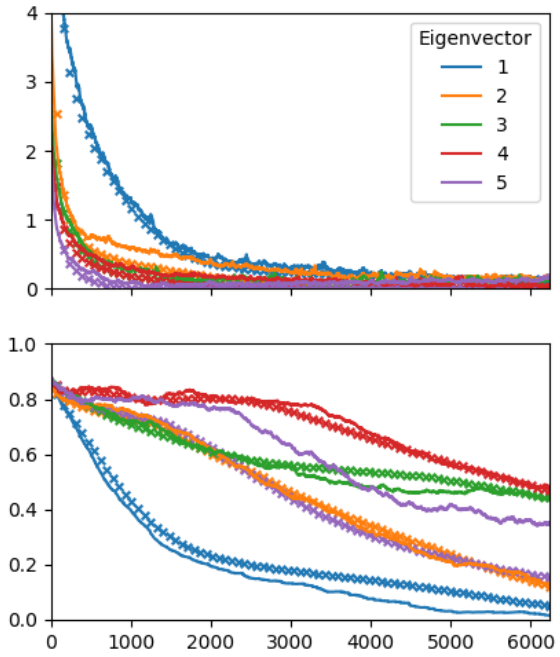

Fig. 1: A comparison of online (solid lines) and batch (x ticks) learning for the eigenvectors of the karate club network graph. The top figure measures the normalized eigenvalue error and the bottom measures cosine distance. Both plots show the number of edge samples along the x-axis. Batch updates are placed 78 samples apart, the number of edges in the graph.

IV. CONCLUSIONS AND FUTURE WORK

We proposed an online learning approach to approximate the eigenvectors of a graph from edge samples. Our method is computationally efficient and demonstrates comparable learning speed to batch gradient descent on our example case.

The majority of the computation in our method stems from the orthogonalization step in the method currently performed using the Grahm-Schmidt process. Reducing the computational complexity of the algorithm depends on improving this step. Additionally, the method is fully inductive and can not generalize to unseen nodes in the graph. There are two approaches to solving this issue, the first is to develop an approach to quickly extend the well converged eigenvector estimates on a portion of the graph to new nodes based on their neighbors. The second approach is a transductive method of learning the values of the eigenvectors as functions of node features so that new nodes could be approximated based on their features.

REFERENCES

[1] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[2] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[3] E. Oja, "Simplified neuron model as a principal component analyzer," *Journal of mathematical biology*, vol. 15, no. 3, pp. 267–273, 1982.

[4] Z. W., "An information flow model for conflict and fission in small groups." *Jounral of Anthropological Research*, vol. 33, pp. 452–473, 1977.