EDGE SAMPLING OF GRAPHS BASED ON EDGE SMOOTHNESS

Kenta Yanagiya[1], *Koki Yamada*[1], *Yasuo Katsuhara*[2], *Tomoya Takatani*[2], *and Yuichi Tanaka*[1]

[1] Tokyo University of Agriculture and Technology
[2] Frontier Research Center, Toyota Motor Corporation

ABSTRACT

Finding important edges in a graph is a crucial problem for various research fields such as network epidemics, signal processing, machine learning, and sensor networks. In this paper, we tackle the problem based on sampling theory on graphs. We convert the original graph to a *line graph* where its nodes and edges, respectively, represent the original edges and the connections between the edges. We then perform node sampling of the line graph based on the *edge smoothness* assumption: This process selects the most important edges in the original graph. We present a general framework of edge sampling based on graph sampling theory and we also reveal a theoretical relationship between the original and line graphs. Experimental results in synthetic graphs validate the effectiveness of our approach against some alternative edge selection methods.

Index Terms— Graph signal processing, edge sampling, edge sparsification

1. INTRODUCTION

Graph signal processing (GSP) is a developing field of signal processing [1–4]. GSP targets *graph signals* whose domain is represented as nodes of a graph. There exist various promising applications of GSP because numerous signals have their underlying structures beyond the standard time and spatial domain relationships, which may be given by a graph. Examples of graph signals include signals on social/brain/transportation/power networks and point clouds [3, 5–12]. GSP aims to extend theories and algorithms for standard signal processing like sampling, filtering, restoration, and compression. In this paper, we focus on a sampling problem of GSP.

Graph signal sampling is a counterpart of that for standard signals [4, 13]. Standard signals are implicitly assumed their (uniform) structure because the sampling period is determined before sampling and is usually fixed throughout the sampling process. In contrast, graph signals have irregular structures. Therefore, sampling methods on graphs have been studied extensively [4–6, 9, 13–17]. However, existing sampling on GSP only considers the sampling of *nodes* as an analogy of that in standard signal processing. There exists another entity to sample exclusive to GSP: *Edges*.

In this paper, we consider a new sampling paradigm for GSP: *Edge sampling*. Edge sampling refers to selecting the most important edges from a given set of edges in the original graph while keeping the original nodes. A new graph is composed of the original nodes and sampled, i.e., selected, edges.

Edge sampling is required in various application fields. For example in network epidemiology, we need to select the most important edges for preventing disease spreading by removing these edges [18, 19]. In the global COVID-19 pandemic, this technique is crucial for policymakers to determine an effective lockdown policy [20]. For network science, we often need to obtain a good abstraction of graphs, i.e., edge sparsification [21–23], for saving computation and storage burden.

These edge sampling methods in different fields share the same purpose in general but their specifications are different. For example, we often need to isolate important nodes in network epidemics, while edge sparsification should keep the characteristics of the original graph like its connectivity. Edge sampling considered in this paper addresses this problem as sampling from a GSP perspective.

In the proposed approach, we assume the *smoothness* of edge weights. That is, the adjacent edge weights, i.e., weights of edges connecting to the same node, are similar. This edge smoothness can be found in various graphs. For example, unweighted graphs have the same edge weights for all edges and thus, they are maximally smooth. We utilize *graph frequency* to measure the smoothness. However, the smoothness in GSP usually refers to the *smoothness of the signal* on the nodes. To properly regard edge smoothness as signal smoothness, we convert the original graph into a *line graph* that represents the edge connection relationship of the original graph. We select nodes in the line graph based on a GSP technique. As a result, selecting important nodes in the line graph is regarded as selecting important edges in the original graph. We can use various effective and high-quality node sampling methods of GSP by converting to the line graph [9, 13, 14]. We also reveal a theoretical relationship between the original and line graphs. In the experiments, our proposed edge sampling method outperforms alternative edge selection methods in terms of the signal reconstruction error.

Notation: A graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of vertices and edges, respectively. The number of vertices is $N = |\mathcal{V}|$ unless otherwise specified. The adjacency matrix of $\mathcal{G}$ is represented as $\mathbf{W}$, where its (m, n)-entry w_{mn} is the edge weight between vertices m and n; $w_{mn} = 0$ for unconnected vertices. The degree matrix $\mathbf{K}$ is diagonal, with mth diagonal element $[\mathbf{K}]_{mm} = k_m := \sum_n w_{mn}$. In this paper, we consider undirected graphs without self-loops, i.e., $[\mathbf{W}]_{nm} = [\mathbf{W}]_{mn}$ and $[\mathbf{W}]_{nn} = 0$ for all m and n. In addition to the (weighted) degree, the number of edges connecting to the node m, i.e., unweighted degree, is defined as $k_m := \sum_n \mathbb{1}_{(\text{if node } m \text{ is connected to node } n)}$.

For an unweighted graph, the incidence matrix $\mathbf{B} \in \mathbb{R}^{N \times |\mathcal{E}|}$ is defined as follows:

$$[\mathbf{B}]_{i\alpha} = \begin{cases} 1 & \text{Edge } \alpha \text{ is incident to vertex } i, \\ 0 & \text{Otherwise.} \end{cases} \tag{1}$$

In other words, each column in $\mathbf{B}$ represents an edge and two nonzero elements in each column correspond to the nodes connecting by the edge. The incidence matrix for a weighted graph $\tilde{\mathbf{B}}$ is similarly defined with replacing 1 in (1) by w_α, where w_α is the weight of the edge α.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

Graph Laplacian is defined as $\mathbf{L} := \mathbf{K} - \mathbf{W}$. Note that $\mathbf{L} = 2\mathbf{K} - \mathbf{B}\mathbf{B}^\top$. Since $\mathbf{L}$ is a real symmetric matrix, it always has an eigendecomposition $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{U} = [\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}]$ is an orthonormal matrix containing the eigenvectors $\mathbf{u}_i$, and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_0, \ldots, \lambda_{N-1})$ consists of the eigenvalues λ_i. These eigenvalues are assumed to be ordered as $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \cdots \leq \lambda_{N-1} = \lambda_{\max}$ without loss of generality. We refer to λ_i as the *graph frequency*. A graph signal $x : \mathcal{V} \rightarrow \mathbb{R}$ is a function that assigns a real value to each node. Graph signals can be written as vectors $\mathbf{x} \in \mathbb{R}^N$ whose nth element, $x[n]$, represents the signal value at the nth node. The graph Fourier transform (GFT) is defined as $\hat{x}[i] = \langle \mathbf{u}_i, \mathbf{x} \rangle = \sum_{n=0}^{N-1} u_i[n]x[n]$.

2. RELATED WORK

In this section, we briefly review existing approaches of sampling theory on graphs and edge sparsification/reduction.

2.1. Graph Signal Sampling

Graph sampling theory is a fundamental topic on GSP [4, 14, 15]. Vertex domain sampling selects samples on $\mathcal{S} \subset \mathcal{V}$ where $\mathcal{S}$ is called a *sampling set*. Since there is no "regular sampling" in general in the graph setting, the sampling quality depends on $\mathcal{S}$ and it has to be determined carefully. Therefore, various sampling strategies have been proposed so far [5, 6, 17]. Similar problems have also been studied in sensor networks and machine learning. In fact, many of them can be regarded as GSP-based sampling set selection [4, 9].

Note that, there have been no edge sampling methods based on graph sampling theory so far because existing methods focus on reducing the number of samples based on the assumption of signal smoothness.

2.2. Edge Sparsification and Reduction

Reduction of edges in a graph has been studied in many fields like graph theory and network epidemiology. There are various methods with different motivations.

In graph theory, the reduction of edges is often called *edge sparsification* [21–23]. The motivation of edge sparsification is to preserve characteristics of the original graph, e.g., eigenvalue distribution and connectivity, in the modified graph. Although there exist theoretical guarantees, edge sparsification methods have three major issues in real applications. First, they often have to modify edge weights, i.e., the edge weights in the sparsified graph are changed. Second, the number of removed edges cannot be determined prior to sparsification. Even under the same parameter(s), the number of removed edges can vary. Third, when a random selection method is considered, the importance of selected edges is not ordered. For adding/removing edges, the whole process of the random selection should be performed again. These are not desirable properties, especially for physical networks.

In network epidemiology, the reduction of edges is utilized to prevent spreading disease by restricting connections between regions, e.g., points-of-interest and cities [18–20]. However, the reduction algorithms are often ad-hoc and their theoretical guarantee is limited.

3. EDGE SAMPLING USING GRAPH SAMPLING THEORY

In this section, we describe our proposed edge sampling method based on graph sampling theory. First, the formulation and overview of our method are presented, then the details of the selection algorithm are discussed.

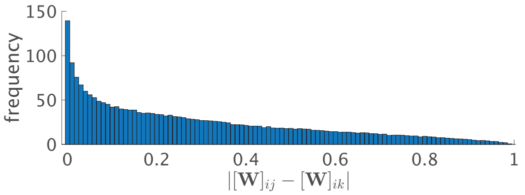


Fig. 1. Histogram of the distribution of the difference between adjacent edge weights, i.e., $|[\mathbf{W}]_{ij} - [\mathbf{W}]_{ik}|$, of random sensor networks ($N = 100$). Average of 100 trials. The horizontal and vertical axes represent the difference of the edge weights and their frequency, respectively.

3.1. Formulation

Suppose that the original graph $\mathcal{G}_0 = (\mathcal{V}, \mathcal{E})$ is given. In general, edge sampling can be represented as the following objective function.

$$\text{find } \mathcal{F} \subset \mathcal{E} \text{ such that } \min f(\mathcal{G}_1), \tag{2}$$

where $\mathcal{G}_1 := (\mathcal{V}, \mathcal{F})$ and $f(\mathcal{G}_1)$ is some cost function. Note that (2) is in general NP-hard. We thus consider to approximately solve (2) with a GSP technique.

In this paper, we consider the following cost function:

$$f(\mathcal{G}_1) := \|\mathbf{w} - \mathrm{Interp}(\mathbf{w}_\mathcal{F} + \mathbf{n})\|_2 \tag{3}$$

where $\mathbf{w} \in \mathbb{R}^{|\mathcal{E}|}$ is a vector composed of the edge weights, i.e., $\mathbf{w} = \mathrm{vec}(\mathbf{W})$, $\mathbf{w}_\mathcal{F}$ is the subset of $\mathbf{w}$ whose elements are specified by $\mathcal{F}$, $\mathbf{n}$ is i.i.d. white Gaussian noise, and Interp is a (linear) interpolation function corresponding to the chosen edge sampling method. That is, (3) leads to finding a good edge subset so that it well estimates removed edge weights under the noisy condition. If the edge weights of the removed edges can be estimated from the remaining ones, the selected edges could represent the original characteristics of the edges. Furthermore, edge weights in a physical measurement are often perturbed during the sensing process, such as sensor networks and biomedical information processing. Therefore, considering the robustness of edge weight perturbation is important.

In this paper, we assume the smoothness of the edge weights in $\mathcal{E}$. This can be formulated as follows:

$$\frac{1}{N} \sum_{i \in \mathcal{V}} \frac{1}{k_i} \sum_{j,k \in \mathcal{N}_i} |[\mathbf{W}]_{ij} - [\mathbf{W}]_{ik}| \leq \epsilon_e, \tag{4}$$

where $\mathcal{N}_i$ is the neighborhood of the vertex i. (4) means the variation of the edge weights for adjacent edges is small. We call this *edge smoothness* in this paper.

Edge smoothness can be verified through unweighted graphs. An unweighted graph has binary edge weights $\{0, 1\}$, and therefore, $|[\mathbf{W}]_{ij} - [\mathbf{W}]_{ik}| = 0$ is always satisfied. The edge smoothness is also approximately satisfied for weighted graphs. Fig. 1 shows the histogram of $|[\mathbf{W}]_{ij} - [\mathbf{W}]_{ik}|$ for random sensor graphs. Edge smoothness refers to that $|[\mathbf{W}]_{ij} - [\mathbf{W}]_{ik}|$ is concentrated at zero. As observed from the figure, the assumption of edge smoothness is reasonable even for weighted graphs.

Smoothness in the vertex domain refers to that the low-frequency energy is dominant in the graph frequency domain: Edge frequency will be a counterpart of edge smoothness. However, there is no

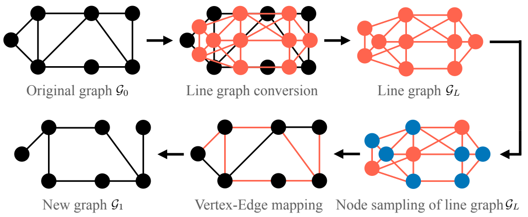

Original graph $\mathcal{G}_0$
Line graph conversion
Line graph $\mathcal{G}_L$
New graph $\mathcal{G}_1$
Vertex-Edge mapping
Node sampling of line graph $\mathcal{G}_L$

Fig. 2. Overview of the proposed method. The original graph is first converted into the line graph (red nodes correspond to the original edges). The important nodes (blue) are then selected from the line graph: This extracts the most important edges in the original graph.

concrete definition for edge frequency. Therefore, we utilize the established definition of *signal smoothness* on graphs. Graph signal smoothness is defined as follows [4, 14, 15],

$$\textstyle\sum_{i=K}^{N-1} \hat{x}^2[i] \leq \epsilon_s, \tag{5}$$

where K is the bandwidth of $\mathbf{x}$. In other words, a smooth graph signal has small energy in its high graph frequency component.

Based on the above assumptions and observations, if we can *flip* roles of the nodes and edges, we can utilize a node sampling method for selecting important edges. In the following, we describe our graph conversion and edge selection method.

3.2. Framework

The overview of the proposed edge sampling method is illustrated in Fig. 2. The main difference between the proposed and existing approaches is that we first convert the original graph $\mathcal{G}_0$ into a *line graph* $\mathcal{G}_L$ [24], and then perform sampling set selection on graphs. A node in the line graph represents an edge in the original graph, and the edge weight of the line graph indicates the relationship between neighboring edges in the original graph. Therefore, *the node selection of the line graph can be regarded as the edge selection of the original graph.*

Since the nodes in the line graph correspond to the original edges, the signal value on the node α can be regarded as the original edge weight of the edge α. We can utilize a fast and efficient sampling set selection method for the line graph to select important edges because we can cast the edge smoothness in (4) into the signal smoothness in (5) through the graph conversion.

The original graph and its converted version have a concrete relationship. We describe this theoretical relationship next.

3.3. Graph Conversion

First of all, we formally define the line graph.

Definition 1. *Suppose that the original graph $\mathcal{G}_0 = (\mathcal{V}, \mathcal{E})$ and its incidence matrices $\mathbf{B}$ and $\tilde{\mathbf{B}}$ are given. The adjacency matrix of the line graph $\tilde{\mathbf{W}}_L \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is defined as follows [24]:*

$$[\tilde{\mathbf{W}}_L]_{\alpha\beta} = \sum_i [\tilde{\mathbf{B}}^\top]_{\alpha i}[\mathbf{B}]_{i\beta}(1-\delta_{\alpha\beta}) = \sum_i [\tilde{\mathbf{B}}^\top]_{\alpha i}[\mathbf{B}]_{i\beta} - 2[\mathbf{C}]_{\alpha\beta}, \tag{6}$$

where α and β are edge indices of the original graph (and therefore, they are node indices in $\mathcal{G}_L$), $\delta_{\alpha\beta}$ is Kronecker delta, and $\mathbf{C} :=$ $\mathrm{diag}(\mathbf{w})$ in which $\mathbf{w} \in \mathbb{R}^{|\mathcal{E}|}$ is the edge weight vector sorted by the edge indices. For unweighted graphs, the line graph is also obtained using (6) *by replacing $\tilde{\mathbf{B}}$ with $\mathbf{B}$.*

Since $\tilde{\mathbf{W}}_L$ is generally directed, we modify it into the undirected line graph by

$$\mathbf{W}_L := \tfrac{1}{2}(\tilde{\mathbf{W}}_L + \tilde{\mathbf{W}}_L^\top). \tag{7}$$

Here, we present the relationship between the degrees of the line graph and the original edge weights.

Proposition 1. *Suppose that the incidence matrix $\mathbf{B}$ or $\tilde{\mathbf{B}}$ of the original graph $\mathcal{G}_0 = (\mathcal{V}, \mathcal{E})$ is given and the adjacency matrix of the line graph is given by* (7)*. Then, the degree of the node α in the line graph, d_α, that corresponds to the edge α connecting the nodes m and n in the original graph is obtained as follows.*

$$d_\alpha = \begin{cases} k_m + k_n - 2 & \textit{for unweighted graphs}, \\ w_\alpha(k_m + k_n - 2) & \textit{for weighted graphs}, \end{cases} \tag{8}$$

where w_α is the weight of the edge α and k_m and k_n are the number of edges connecting to the node m and n, respectively.

Its proof is omitted due to limitation of space.

Proposition 1 indicates that the high-degree nodes in the line graph correspond to the original edges connecting high-degree nodes. In sampling set selection based on signal smoothness, selected nodes often have high-degree. Hence, selecting important nodes in the line graph can be regarded as selecting important edges in the original graph.

3.4. Node Sampling of Line Graph

As previously mentioned, we can use an arbitrary sampling set selection for the line graph. The important property of GSP-based sampling set selection methods is that many of them are designed to be robust to noise [4]. This satisfies the requirement of (3).

While any sampling set selection algorithm can be utilized, two issues should be considered according to applications. The first property is the characteristics of selected nodes in the line graph. Distributed selection algorithms select important nodes such that they are uniformly distributed in space. This is beneficial for edge sparsification. In contrast, concentrated selection algorithms do not have such a restriction on the node distribution: The selected nodes are often concentrated. This could be utilized for network epidemics.

The second property is computation complexity. The number of the nodes in the line graph is $|\mathcal{E}|$, which is often greater than that of the original graph N. Hence, fast selection methods are preferred especially for edge sampling: We recommend using a fast selection method like [9].

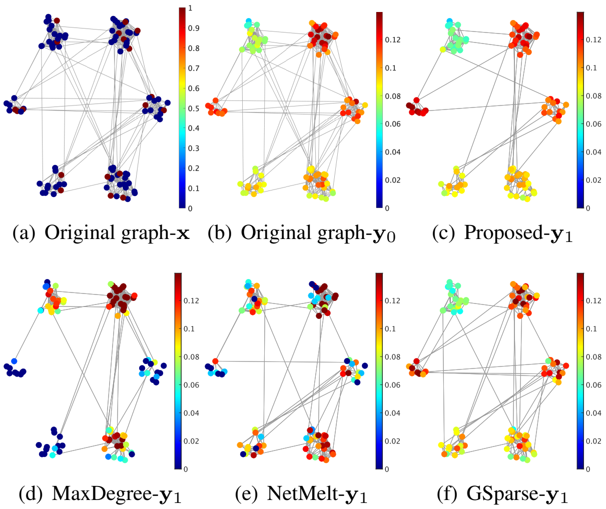
4. EXPERIMENTS

4.1. Setup

In this paper, the proposed edge sampling method is applied for graph sparsification. The accuracy of the sparsification is compared in two measures: 1) Edge weight reconstruction error and 2) MSE of diffused random signals.

Edge weight reconstruction error: We recover removed edge weights with a graph signal reconstruction method based on the bandlimited assumption [9] and they are computed by (3). If $\mathcal{F}$ in $\mathcal{G}_1$ is a good abstraction of $\mathcal{E}$ in $\mathcal{G}_0$, the recovered edge weights should be close to the original ones.

MSE of diffused random signals: The MSE of diffused random signals are computed as follows. Let $\mathbf{L}_0$ be the graph Laplacian of $\mathcal{G}_0$ and $\mathbf{L}_1$ be that of $\mathcal{G}_1$. The diffused signal on $\mathcal{G}_k$ ($k \in \{0,1\}$) is represented as $\mathbf{y}_k := h(\mathbf{L}_k)\mathbf{x} = \mathbf{U}_k h(\mathbf{\Lambda}_k)\mathbf{U}_k^\top \mathbf{x}$, where $\mathbf{x} \in \{0,1\}^N$



(a) Original graph-$\mathbf{x}$ (b) Original graph-$\mathbf{y}_0$ (c) Proposed-$\mathbf{y}_1$

(d) MaxDegree-$\mathbf{y}_1$ (e) NetMelt-$\mathbf{y}_1$ (f) GSparse-$\mathbf{y}_1$

Fig. 3. Graph sparsification and diffusion example: Community graph. Diffused signals are also shown.

is the input signal, $h(\mathbf{L}_k)$ is the lowpass graph filter on $\mathcal{G}_k$, and $\mathbf{L}_k := \mathbf{U}_k\mathbf{\Lambda}_k\mathbf{U}_k^\top$. We set $h(\lambda) = e^{-t\lambda}$ where $t > 0$ is a parameter. Nonzero elements in $\mathbf{x}$ are randomly chosen and their number is set to be 20. Finally, the MSE of the diffused signal is calculated by $\text{MSE}(\mathbf{y}_0, \mathbf{y}_1) = \frac{1}{N}\|\mathbf{y}_0 - \mathbf{y}_1\|_2^2$. If $\mathcal{G}_1$ preserves the original structure, the diffused signal values on $\mathcal{G}_1$ will become similar to those on $\mathcal{G}_0$. This results in a low MSE.

In the experiment, we use following weighted graphs with $N = 100$: 1) community graph: $|\mathcal{E}| = 665$, 2) random sphere graph: $|\mathcal{E}| = 588$, and 3) random graph based on Erdős–Rényi model: $|\mathcal{E}| = 906$. Edge weights are randomly drawn from a truncated normal distribution where the mean is set to 1 and the minimum and maximum weights are 0.75 and 1.25, respectively.

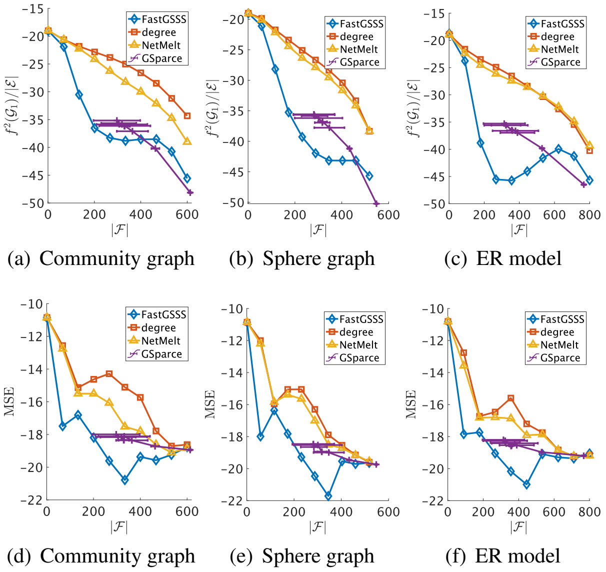
In the proposed edge sampling method, we use FastGSSS [9] as a sampling set selection of the line graph. The performance is compared with the following edge sampling methods:

- MaxDegree (deterministic): Greedy selection. Edges having the largest $k_m + k_n$ are selected one-by-one.
- NetMelt (deterministic) [25]: Edge selection based on the score calculated from the eigenvectors of $\mathbf{L}_0$.
- GSparse (random) [21]: Graph sparsification based on effective resistances, where a random selection of edges is performed based on a probability proportional to the effective resistance of $\mathcal{G}_0$.

The proposed method and the first two alternative methods are deterministic approaches: The number of edges is specified prior to edge sampling and the edges to be removed are fixed as long as the graph is fixed. In contrast, [21] is a random approach that requires a sparsification parameter between $\frac{1}{\sqrt{N}}$ and 1. In other words, the number of removed edges cannot be determined in the random method. Furthermore, even under the same parameter, the removed edge positions are changed in each realization and the number of them can vary due to random selection with replacement.

4.2. Experimental Results

Fig. 3 shows the sparsified graphs as well as the diffused signals on them. It is clear that the proposed method and GSparse are almost connected while MaxDegree and NetMelt often isolate nodes. It is also observed that the diffused signals of Figs. 3(b), (c), and (f) are similar to each other.



(a) Community graph (b) Sphere graph (c) ER model

(d) Community graph (e) Sphere graph (f) ER model

Fig. 4. Comparison of objective performances of sparsified graphs. (a)–(c): Normalized edge weight reconstruction errors. (d)–(f): MSEs of diffused signals in dB. Averaged results after 100 runs are shown. The horizontal lines of GSparse denote the variations of $|\mathcal{F}|$ (i.e., the minimum/maximum number of edges) in the experiment.

Figs. 4(a)–(c) show $f(\mathcal{G}_1)$ in (3) as functions of $|\mathcal{F}|$. As previously mentioned, the number of removed edges by GSparse varies even under the same parameter. Therefore, we also illustrate such a variation in the figure. The proposed method shows consistently lower edge weight reconstruction errors than the other methods. It is observed that $f(\mathcal{G}_1)$ of the proposed method oscillates. This is because we use one set of parameters for edge sampling of the graph. This could be modified by using the optimal parameter set for each $|\mathcal{F}|$: This is left for future work.

Figs. 4(d)–(f) show the average MSEs of the diffused signal according to $|\mathcal{F}|$ in the sparsified graph. As observed in the sparsified graphs, the proposed method and GSparse present comparable MSEs and they are better than MaxDegree and NetMelt. The proposed method enables a wide range of edge sparsification factors because it can specify the number of edges thanks to deterministic sampling. In contrast, GSparse has a small admissible range of $|\mathcal{E}|$. As previously mentioned, the number of removed edges by GSparse significantly varies under the same parameter, where its range sometimes exceeds 200 which is about one-third of $|\mathcal{E}|$.

5. CONCLUSION

In this paper, we propose an edge sampling method based on graph sampling theory. We first convert the original graph into a line graph, and perform a sampling set selection of graphs. The edge smoothness characteristic is converted as the signal smoothness via the graph conversion. The experimental results on edge sparsification reveal the effectiveness of the proposed method against some alternative methods.

6. REFERENCES

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

[2] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[3] G. Cheung, E. Magli, Y. Tanaka, and M. Ng, “Graph spectral image processing,” *Proc. IEEE*, vol. 106, no. 5, pp. 907–930, May 2018.

[4] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, “Sampling signals on graphs: From theory to applications,” *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 14–30, 2020.

[5] Y. Tanaka, “Spectral domain sampling of graph signals,” *IEEE Trans. Signal Process.*, vol. 66, no. 14, pp. 3752–3767, Jul. 2018.

[6] Y. Tanaka and Y. C. Eldar, “Generalized sampling on graphs with subspace and smoothness priors,” *IEEE Trans. Signal Process.*, vol. 68, pp. 2272–2286, 2020.

[7] H. E. Egilmez, E. Pavez, and A. Ortega, “Graph learning from data under laplacian and structural constraints,” *IEEE J. Sel.Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, Sep. 2017.

[8] K. Yamada, Y. Tanaka, and A. Ortega, “Time-varying graph learning with constraints on graph temporal variation,” *arXiv preprint arXiv:2001.03346*, 2020.

[9] A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega, “Eigendecomposition-Free Sampling Set Selection for Graph Signals,” *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2679–2692, May 2019.

[10] Y. Tanaka and A. Sakiyama, “M-channel oversampled graph filter banks,” *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3578–3590, Jul. 2014.

[11] A. Sakiyama and Y. Tanaka, “Oversampled graph Laplacian matrix for graph filter banks,” *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6425–6437, Dec. 2014.

[12] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, “Graph signal denoising via trilateral filter on graph spectral domain,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 137–148, Jun. 2016.

[13] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, “Signals on Graphs: Uncertainty Principle and Sampling,” *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4845–4860, Sep. 2016.

[14] A. Anis, A. Gadde, and A. Ortega, “Efficient sampling set selection for bandlimited graph signals using graph spectral proxies,” *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, Jul. 2016.

[15] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, “Discrete signal processing on graphs: Sampling theory,” *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

[16] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Sampling of graph signals with successive local aggregations,” *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1832–1843, 2016.

[17] J. Hara, Y. Tanaka, and Y. C. Eldar, “Generalized graph spectral sampling with stochastic priors,” in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.,*, 2020.

[18] C. Nowzari, V. M. Preciado, and G. J. Pappas, “Analysis and Control of Epidemics: A Survey of Spreading Processes on Complex Networks,” *IEEE Control Systems Magazine*, vol. 36, no. 1, pp. 26–46, Feb. 2016.

[19] A. Nishi, G. Dewey, A. Endo, S. Neman, S. K. Iwamoto, M. Y. Ni, Y. Tsugawa, G. Iosifidis, J. D. Smith, and S. D. Young, “Network interventions for managing the COVID-19 pandemic and sustaining economy,” *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 285–30 294, Dec. 2020.

[20] S. Chang, E. Pierson, P. W. Koh, J. Gerardin, B. Redbird, D. Grusky, and J. Leskovec, “Mobility network models of COVID-19 explain inequities and inform reopening,” *Nature*, vol. 589, no. 7840, pp. 82–87, Jan. 2021.

[21] D. A. Spielman and N. Srivastava, “Graph Sparsification by Effective Resistances,” *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1913–1926, Jan. 2011.

[22] D. A. Spielman and S.-H. Teng, “Spectral sparsification of graphs,” *SIAM Journal on Computing*, vol. 40, no. 4, pp. 981–1025, 2011.

[23] W.-S. Fung, R. Hariharan, N. J. A. Harvey, and D. Panigrahi, “A General Framework for Graph Sparsification,” *SIAM Journal on Computing*, vol. 48, no. 4, pp. 1196–1223, Jan. 2019.

[24] T. S. Evans and R. Lambiotte, “Line Graphs of Weighted Networks for Overlapping Communities,” *The European Physical Journal B*, vol. 77, no. 2, pp. 265–272, Sep. 2010.

[25] C. Chen, H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, “Eigen-Optimization on Large Graphs by Edge Manipulation,” *ACM Transactions on Knowledge Discovery from Data*, vol. 10, no. 4, pp. 49:1–49:30, Jun. 2016.