Online Edge Flow Prediction Over Expanding Simplicial Complexes

Yang, Maosheng; Das, Bishwadeep; Isufi, Elvin

DOI
10.1109/ICASSP49357.2023.10096364

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096364

ONLINE EDGE FLOW PREDICTION OVER EXPANDING SIMPLICIAL COMPLEXES

Maosheng Yang, Bishwadeep Das and Elvin Isufi

ABSTRACT

Simplicial convolutional filters can process signals defined over levels of a simplicial complex such as nodes, edges, triangles, and so on with applications in e.g., flow prediction in transportation or financial networks. However, the underlying topology expands over time in a way that new edges and triangles form. For example, in a transportation network, a new connection between two locations is newly built, or in a currency exchange market, two currencies can be exchanged without an intermediate currency that can be understood as a new edge between them. To handle the streaming nature of data, we propose an online prediction for edge flows which generalizes to other higher-order simplicial signals. This is achieved by updating the filter coefficients via an online gradient descent with a provable sub-linear regret relative to the simplicial filter optimized over the whole sequence of edge flows. The update of the filter coefficients associated with the lower and upper Hodge Laplacians can be uncoupled in general. We test the online edge flow prediction on an expanding synthetic simplicial complex and a coauthorship complex showing a close performance to the offline counterpart.

1. INTRODUCTION

Graph signal processing tools have become ubiquitous for manipulating graph signals. However, they are limited in the node signal space [1, 2] whereas we often encounter signals that are naturally associated with edges, e.g., blood flow between different areas in the brain [3], water flow in a water distribution network, data flow in a communication network, or traffic flow in a road network [4, 5]. We typically model these signals as flows over the edges of a network, which also has applications in modelling exchange rates in financial markets [6], representing user preferences in statistical ranking [6], or analyzing games or politics [7, 8, 9].

Analogous to utilizing edges for modelling the pairwise relationship between nodes when processing graph signals, we exploit the relationships between edges through a common incidence node or a common triangle where they participate to form a triadic relation when processing edge flows [10]. Simplicial complexes are a generalization of graphs that can model the above two edge adjacencies as lower and upper adjacencies with favourable algebraic properties [11]. By associating subsets of nodes with signals, we obtain simplicial signals, where edge flows are an instance for a pair of nodes. Recent works have established a framework to analyze and process such signals from both spatial/simplicial and spectral perspectives, including signal reconstruction methods [12, 13], Fourier transforms [10], simplicial filters [10, 14].

In this paper, we are interested in predicting edge flows sequentially when new edges are forming over time, i.e., the underlying simplicial complex expands over time [15, 16, 17]. This arises in numerous applications, for instance, predicting the water flow in a new pipe between hydraulic components of a water network [18], providing a global ranking where new alternatives are coming over time in statistical ranking, or predicting the exchange rates in a financial market where a new trade between two commodities establishes over time [6, 10]. Nevertheless, existing methods [19, 13, 20] only process simplicial signals on static settings and do not capture the expanding nature of the topology.

To account for the underlying topological dynamics, we propose an online framework to perform edge flow prediction over an expanding simplicial complex. In detail, we first use the simplicial convolutional filter proposed in [10, 21] as the flow predictor on the new edge; then we consider an online gradient descent algorithm to update the filter parameters [22, 23], generalizing the methods developed for the node signal space [24, 25, 26]. Online gradient descent, as a simple update rule, has sub-linear regret bounds [22]. We show that both the prediction and update steps enjoy low computational costs suitable for the streaming nature of the edge addition. We also uncouple the filter update into the two sets of updates associated with the lower and upper edge adjacencies, reducing the computation cost. Lastly, we evaluate this algorithm with both synthetic and real-world data.

2. SIMPLICIAL SIGNALS AND SIMPLICIAL FILTERS

A k-simplex s^k is a set of $k+1$ vertices. For a k-simplex, its faces are all its subsets with k vertices and its cofaces are all $(k+1)$-simplices that have s^k as a face. A simplicial complex (SC) $\mathcal{X}$ is a collection of simplices which are closed under inclusion, i.e., if a simplex is in the SC, then its subsets are also in the SC. Geometrically, a node is a 0-simplex, an edge is a 1-simplex, and a filled triangle is a 2-simplex. A graph with a node and edge set is a simple SC [11, 13].

Given a SC $\mathcal{X}$ over a set of nodes $\mathcal{V} = \{1, 2, \ldots, |\mathcal{V}|\}$, with the edge set $\mathcal{E}$ and the triangle set $\mathcal{P}$, we assign each edge $e = \{i, j\}$ and triangle $p = \{i, j, k\}$ an orientation according to the lexicographical ordering of their vertices, denoted by $e = [i, j], i < j$ and $p = [i, j, k], i < j < k$. We can use the incidence matrix $\mathbf{B}_1 \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$ to describe the relationship between nodes and edges, whose eth column has nonzero entries when node i is incident to edge e, i.e., $[\mathbf{B}_1]_{i,e} = -1$ for $e = [i, \cdot]$ and $[\mathbf{B}_1]_{i,e} = 1$ for $e = [\cdot, i]$. Likewise, the incidence matrix $\mathbf{B}_2 \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{P}|}$ describes the relationship between edges and triangles, whose pth column has nonzero entries when edge e participates in the triangle p, i.e., $[\mathbf{B}_2]_{e,p} = -1$ for $e = [i, j], p = [i, \cdot, j]$, and $[\mathbf{B}_2]_{e,p} = 1$ for $e = [i, j], p = [i, j, \cdot]$ or $p = [\cdot, i, j]$.

Moreover, a Hodge Laplacian in the edge space can be further defined as $\mathbf{L}_1 = \mathbf{B}_1^\top \mathbf{B}_1 + \mathbf{B}_2 \mathbf{B}_2^\top$ where the lower Laplacian $\mathbf{L}_\mathrm{d} := \mathbf{B}_1^\top \mathbf{B}_1$ describes the edge-to-edge lower adjacencies via the common nodes, while the upper Laplacian $\mathbf{L}_\mathrm{u} := \mathbf{B}_2 \mathbf{B}_2^\top$ describes the edge-to-edge upper adjacencies via the common triangles.

The authors are from faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. The work of Maosheng Yang was supported by the TU Delft AI Labs Programme. e-mails: {m.yang-2, b.das, e.isufi-1}@tudelft.nl.

Given a SC, we can define a k-simplicial signal space by associating the set of k-simplices with a real vector space. For instance, we can define an edge flow $\mathbf{f}: \mathcal{E} \to \mathbb{R}^{|\mathcal{E}|}$, a node signal $\mathbf{v}: \mathcal{V} \to \mathbb{R}^{|\mathcal{V}|}$, and a triangle flow $\mathbf{p}: \mathcal{P} \to \mathbb{R}^{|\mathcal{P}|}$. If the flow value on edge e follows $[\mathbf{f}]_e > 0$, the flow orientation is aligned with the orientation of edge e, and the flow is opposite, otherwise.

Given the Hodge Laplacian $\mathbf{L}_1$ and its lower and upper counterparts, a simplicial convolutional filter is defined as

$$\mathbf{H} := \mathbf{H}(\boldsymbol{\alpha}, \boldsymbol{\beta}; \mathbf{L}_\mathrm{d}, \mathbf{L}_\mathrm{u}) = \sum_{k=0}^{K_\mathrm{d}} \alpha_k \mathbf{L}_\mathrm{d}^k + \sum_{k=0}^{K_\mathrm{u}} \beta_k \mathbf{L}_\mathrm{u}^k \tag{1}$$

which is a matrix polynomial of the lower and upper Laplacians $\mathbf{L}_\mathrm{d}$ and $\mathbf{L}_\mathrm{u}$ with vectors $\boldsymbol{\alpha} = [\alpha_0, \dots, \alpha_{K_\mathrm{d}}]^\top \in \mathbb{R}^{K_\mathrm{d}+1}$ and $\boldsymbol{\beta} = [\beta_0, \dots, \beta_{K_\mathrm{u}}]^\top \in \mathbb{R}^{K_\mathrm{u}+1}$ collecting the filter coefficients with filter orders K_d and K_u, respectively. The filtering of an edge flow $\mathbf{f}$ with a simplicial convolutional filter $\mathbf{H}$ is a shift-and-sum operation:

- *shift*: the filter first performs K_d consecutive lower simplicial shifts in terms of the lower adjacencies to obtain $\mathbf{f}_\mathrm{d}^{(1)}, \dots, \mathbf{f}_\mathrm{d}^{(K_\mathrm{d})}$ with $\mathbf{f}_\mathrm{d}^{(k)} := \mathbf{L}_\mathrm{d}^k \mathbf{f}$ denoting the k-th lower shift result; and likewise, K_u consecutive upper shifts in terms of the upper adjacencies to obtain $\mathbf{f}_\mathrm{u}^{(1)}, \dots, \mathbf{f}_\mathrm{u}^{(K_\mathrm{u})}$ with $\mathbf{f}_\mathrm{u}^{(k)} := \mathbf{L}_\mathrm{u}^k \mathbf{f}$ denoting the k-th upper shift result;
- *sum*: the filter performs a weighted linear combination of the above shifted results as $(\alpha_0 + \beta_0)\mathbf{f} + \sum_{k=1}^{K_\mathrm{d}} \alpha_k \mathbf{f}_\mathrm{d}^{(k)} + \sum_{k=1}^{K_\mathrm{u}} \beta_k \mathbf{f}_\mathrm{u}^{(k)}$.

More importantly, the shift operation is local and admits a distributed recursion. In particular, the one-step lower shift on an edge e is a linear combination of the edge flows in its lower neighborhood $\mathcal{N}_{\mathrm{d},e}$, which share a common node with edge e, given by $[\mathbf{L}_\mathrm{d}\mathbf{f}]_e = \sum_{j \in \mathcal{N}_{\mathrm{d},e} \cup \{e\}} [\mathbf{L}_\mathrm{d}]_{e,j} \mathbf{f}_j$. Likewise, the upper shift is a local operation happening in the upper neighborhood $\mathcal{N}_{\mathrm{u},e}$, which share a common triangle with edge e, given by $[\mathbf{L}_\mathrm{u}\mathbf{f}]_e = \sum_{j \in \mathcal{N}_{\mathrm{u},e} \cup \{e\}} [\mathbf{L}_\mathrm{u}]_{e,j} \mathbf{f}_j$. Furthermore, the k-step shift can be obtained via a one-step shift of the $(k-1)$-shifted result, i.e., $\mathbf{f}_\mathrm{d}^{(k)} = \mathbf{L}_\mathrm{d} \mathbf{f}_\mathrm{d}^{(k-1)}$, and $\mathbf{f}_\mathrm{u}^{(k)} = \mathbf{L}_\mathrm{u} \mathbf{f}_\mathrm{u}^{(k-1)}$. This further allows a recursive implementation of the filter. Thus, when computing the output for an edge e, it has a total communication cost of order $\mathcal{O}(|\mathcal{N}_{\mathrm{d},e}|K_\mathrm{d} + |\mathcal{N}_{\mathrm{u},e}|K_\mathrm{u})$ [10].

3. ONLINE EDGE FLOW PREDICTION OVER EXPANDING SIMPLICIAL COMPLEXES

We apply the simplicial filter (1) to predict edge flows in an online fashion over an expanding simplicial complex, where we assume new edges are forming over time. In what follows, we add a subscript t to the previously introduced variables to denote their counterparts at time instance t.

In a SC with a fixed node set $\mathcal{V}$, we have a stream of edge formation over time, i.e., edge sets $\mathcal{E}_0, \mathcal{E}_1, \dots, \mathcal{E}_t$, which leads to a stream of growing triangles, i.e., triangle sets $\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_t$. Suppose a new edge e_t is formed at time t, which is adjacent to its lower neighbors via a node, and this edge e_t together with two other existing edges forms a new triangle p_t, without loss of generality. We account for both the newly formed lower and upper adjacencies by adding columns $\mathbf{b}_{1,t} \in \mathbb{R}^{|\mathcal{V}|}$ and $\mathbf{b}_{2,t} \in \mathbb{R}^{|\mathcal{E}_t|}$ to the incidence matrices, $\mathbf{B}_{1,t-1} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}_t|}$ and $\mathbf{B}_{2,t-1} \in \mathbb{R}^{|\mathcal{E}_t| \times |\mathcal{P}_t|}$, respectively. The updated incidence matrices read as

$$\mathbf{B}_{1,t} = \begin{bmatrix} \mathbf{B}_{1,t-1} & \mathbf{b}_{1,t} \end{bmatrix}, \quad \mathbf{B}_{2,t} = \left[\begin{array}{c|c} \mathbf{B}_{2,t-1} & \multirow{2}{*}{$\mathbf{b}_{2,t}$} \\ 0 & \end{array}\right] \tag{2}$$

with $|\mathcal{E}_t| = |\mathcal{E}_{t-1}| + 1$, and $|\mathcal{P}_t| = |\mathcal{P}_{t-1}| + 1$; see Fig. 1 for an illustration of $\mathbf{b}_{1,t}$ and $\mathbf{b}_{2,t}$. We then update the Hodge Laplacians from the previous ones $\mathbf{L}_{\mathrm{d},t-1}$ and $\mathbf{L}_{\mathrm{u},t-1}$ as

$$\mathbf{L}_{\mathrm{d},t} = \begin{bmatrix} \mathbf{L}_{\mathrm{d},t-1} & \mathbf{B}_{1,t-1}^\top \mathbf{b}_{1,t} \\ \mathbf{b}_{1,t}^\top \mathbf{B}_{1,t-1} & \mathbf{b}_{1,t}^\top \mathbf{b}_{1,t} \end{bmatrix} \tag{3}$$

and

$$\mathbf{L}_{\mathrm{u},t} = \begin{bmatrix} \mathbf{L}_{\mathrm{u},t-1} & \overline{\mathbf{B}}_{2,t-1}^\top \mathbf{b}_{2,t} \\ \mathbf{b}_{2,t}^\top \overline{\mathbf{B}}_{2,t-1} & \mathbf{b}_{2,t}^\top \mathbf{b}_{2,t} \end{bmatrix} \tag{4}$$

with $\overline{\mathbf{B}}_{2,t-1} := \begin{bmatrix} \mathbf{B}_{2,t-1} & 0 \end{bmatrix} \in \mathbb{R}^{|\mathcal{E}_{t-1}| \times |\mathcal{P}_t|}$. Note that a new triangle is not necessarily formed, in which case the update of $\mathbf{B}_{2,t}$ and $\mathbf{L}_{\mathrm{u},t}$ is not needed. Given the expanding SC model, we consider the following two-step online edge flow prediction at time t: first, an edge flow prediction step, and second, an online filter update step.

Edge flow prediction. Consider a sequence of edge flows $\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_T$ over time $t = 0, 1, \dots, T$, which are of different dimensions $|\mathcal{E}_0|, |\mathcal{E}_1|, \dots, |\mathcal{E}_T|$, due to the edge expansion of the SC. Our goal is to predict the edge flow $[\mathbf{f}_t]_{e_t}$ on the newly formed edge e_t at time t based on the existing edge flow $\mathbf{f}_{t-1}$, the current predictor parameters, and the newly updated SC topology.

To perform the prediction of $[\mathbf{f}_t]_{e_t}$, we consider the prediction model with the input edge flow $\mathbf{f}_{t-1}$ from the previous time instance via a simplicial filter $\mathbf{H}_t := \mathbf{H}_t(\mathbf{h}_{t-1}; \mathbf{L}_{\mathrm{d},t}, \mathbf{L}_{\mathrm{u},t})$

$$\hat{\mathbf{f}}_t = \mathbf{H}_t \begin{bmatrix} \mathbf{f}_{t-1} \\ g \end{bmatrix} = \left(\sum_{k=0}^{K_\mathrm{d}} \alpha_{k,t-1} \mathbf{L}_{\mathrm{d},t}^k + \sum_{k=0}^{K_\mathrm{u}} \beta_{k,t-1} \mathbf{L}_{\mathrm{u},t}^k \right) \begin{bmatrix} \mathbf{f}_{t-1} \\ g \end{bmatrix}$$

where we emphasise two points: 1) we build the filter $\mathbf{H}_t$ from the filter coefficients $\mathbf{h}_{t-1} = [\boldsymbol{\alpha}_{t-1}^\top, \boldsymbol{\beta}_{t-1}^\top]^\top$ at the previous time instance $t-1$ with the updated Hodge Laplacians $\mathbf{L}_{\mathrm{d},t}$ and $\mathbf{L}_{\mathrm{u},t}$ [cf. (3) and (4)], and 2) we assume the edge flow on the new edge has a default value g, e.g., zero, or the mean or median of the edge flow at $t-1$. Then, the predicted flow $[\hat{\mathbf{f}}_t]_{e_t}$ on edge e_t can be equivalently written as

$$[\hat{\mathbf{f}}_t]_{e_t} = [\mathbf{F}_t \mathbf{h}_{t-1}]_{e_t} = [\mathbf{F}_t]_{e_t,:} \mathbf{h}_{t-1} \tag{5}$$

where $[\mathbf{F}_t]_{e_t,:}$ is the e_tth row of $\mathbf{F}_t$, which is given by

$$\mathbf{F}_t = [\bar{\mathbf{f}}_t \ \bar{\mathbf{f}}_{\mathrm{d},t}^{(1)} \ \dots \ \bar{\mathbf{f}}_{\mathrm{d},t}^{(K_\mathrm{d})} \ \bar{\mathbf{f}}_t \ \bar{\mathbf{f}}_{\mathrm{u},t}^{(1)} \ \dots \ \bar{\mathbf{f}}_{\mathrm{u},t}^{(K_\mathrm{u})}] \tag{6}$$

with $\bar{\mathbf{f}}_t := [\mathbf{f}_{t-1}^\top, g]^\top$ and $\bar{\mathbf{f}}_{\mathrm{d},t}^{(k)} := \mathbf{L}_{\mathrm{d},t}^k \bar{\mathbf{f}}_t$ and $\bar{\mathbf{f}}_{\mathrm{u},t}^{(k)} := \mathbf{L}_{\mathrm{u},t}^k \bar{\mathbf{f}}_t$ denoting the shifted results by the updated Laplacians. Moreover, the prediction (5) can be written w.r.t. $\boldsymbol{\alpha}_{t-1}$ and $\boldsymbol{\beta}_{t-1}$ as

$$[\hat{\mathbf{f}}_t]_{e_t} = [\mathbf{F}_{\mathrm{d},t}]_{e_t,:} \boldsymbol{\alpha}_{t-1} + [\mathbf{F}_{\mathrm{u},t}]_{e_t,:} \boldsymbol{\beta}_{t-1} \tag{7}$$

where we collect the lower shifted flows and the upper shifted flows separately, i.e., define

$$\mathbf{F}_{\mathrm{d},t} := [\bar{\mathbf{f}}_t \ \bar{\mathbf{f}}_{\mathrm{d},t}^{(1)} \ \dots \ \bar{\mathbf{f}}_{\mathrm{d},t}^{(K_\mathrm{d})}] \text{ and } \mathbf{F}_{\mathrm{u},t} := [\bar{\mathbf{f}}_t \ \bar{\mathbf{f}}_{\mathrm{u},t}^{(1)} \ \dots \ \bar{\mathbf{f}}_{\mathrm{u},t}^{(K_\mathrm{u})}].$$

The entries of row $[\mathbf{F}_t]_{e_t,:}$ can be computed locally over the lower and upper neighborhood of e_t and they admit a recursive implementation via $[\bar{\mathbf{f}}_{\mathrm{d},t}^{(k)}]_{e_t} = [\mathbf{L}_{\mathrm{d},t} \bar{\mathbf{f}}_t^{(k-1)}]_{e_t}$ which is given by

$$\sum_{j \in \mathcal{N}_{\mathrm{d},e_t} \cup \{e_t\}} [\mathbf{L}_{\mathrm{d},t}]_{e_t,j} [\bar{\mathbf{f}}_t^{(k-1)}]_j = \sum_{j \in \mathcal{N}_{\mathrm{d},e_t}} [\mathbf{L}_{\mathrm{d},t}]_{e_t,j} [\mathbf{f}_{t-1}^{(k-1)}]_j$$

and likewise for the upper shifted entries. In the distributed implementation, the communication complexity of computing $[\mathbf{F}_t]_{e_t,:}$ is of order $\mathcal{O}(|\mathcal{N}_{d,e_t}|K_\mathrm{d} + |\mathcal{N}_{u,e_t}|K_\mathrm{u})$, yielding from the K_d lower

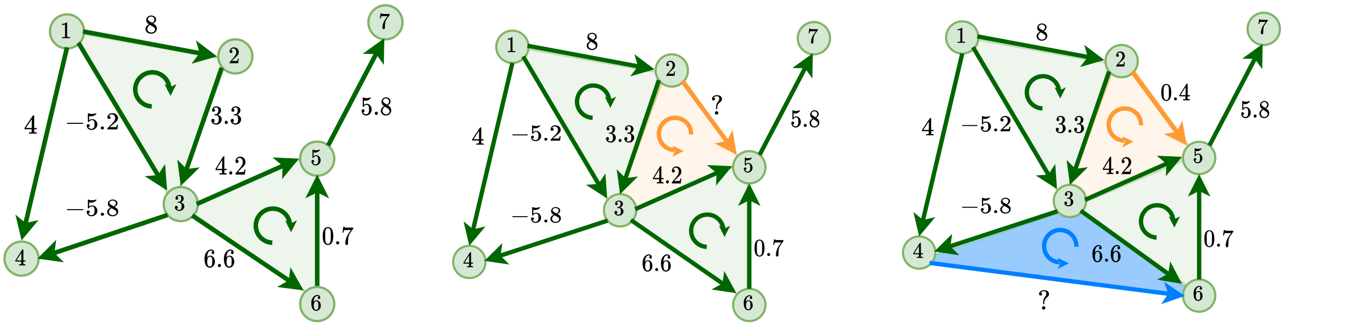

Fig. 1. Online edge flow prediction at time instance t over an expanding simplicial complex which includes nodes, edges and filled triangles (shaded area). The arrows on the edges and the arcs with an arrow on the triangles indicate the reference orientations of the edges and triangles, respectively. The open triangle $\{1,3,4\}$ is not a 2-simplex. *(Left)*: the edge flow $\mathbf{f}_{t-1}$ and the SC at time instance $t-1$. *(Middle)*: at time instance t, a new edge $e_t=[2,5]$ is formed, which creates new lower adjacency via nodes 2 and 5 and upper adjacency via triangle $[2,3,5]$, i.e., we have $[\mathbf{b}_1]_2=-1$, $[\mathbf{b}_1]_5=1$, and $[\mathbf{b}_2]_{[2,3]}=1$, $[\mathbf{b}_2]_{[2,5]}=-1$, $[\mathbf{b}_2]_{[3,5]}=1$. Then, the edge flow is predicted by $[\hat{\mathbf{f}}_t]_{e_t}=[\mathbf{H}_t\bar{\mathbf{f}}_t]_{e_t}$ [cf. (5) and (7)]. *(Right)*: When the edge flow on $e_{t+1}=[2,5]$ is know, the filter coefficients $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ are updated [cf. (11) and (14)]; Meanwhile, a new edge $e_{t+1}=[4,6]$ is formed at $t+1$, the edge flow on which is predicted again based on (5) again.

and K_{u} upper consecutive shifts, and an inner product with a cost of order $\mathcal{O}(K_{\mathrm{d}}^2+K_{\mathrm{u}}^2)$. With the edge flow $[\hat{\mathbf{f}}_t]_{e_t}$ on the new edge e_t predicted, we update the simplicial filter coefficients.

Online filter learning. After the true edge flow $[\mathbf{f}_t]_{e_t}$ is revealed, we update the filter coefficient $\mathbf{h}_t$. In an offline setting, $\mathbf{h}_t$ can be obtained by minimizing the accumulated loss up to time t

$$\mathbf{h}_t=\underset{\mathbf{h}}{\operatorname{argmin}}\sum_{\tau=0}^{t}\ell_\tau(\mathbf{h}) \tag{8}$$

with the loss at time τ defined as, $\ell_\tau(\mathbf{h})=([\mathbf{F}_\tau]_{e_\tau,:}\mathbf{h}-[\mathbf{f}_\tau]_{e_\tau})^2+\mu\|\mathbf{h}\|_2^2$. The solution to (8) is also referred to as a batch solution, which may be computationally demanding to obtain for every t.

In the online learning setting, however, we update the filter coefficients $\mathbf{h}_t$ based on the instantaneous loss at time instance t,

$$\ell_t(\mathbf{h})=([\mathbf{F}_t]_{e_t,:}\mathbf{h}-[\mathbf{f}_t]_{e_t})^2+\mu\|\mathbf{h}\|_2^2, \tag{9}$$

which measures the ℓ_2-norm distance of the predicted edge flow $[\hat{\mathbf{f}}_t]_{e_t}$ and the true one $[\mathbf{f}_t]_{e_t}$ with a regularizer $\|\mathbf{h}\|_2^2$ weighted by $\mu>0$. The loss in (9) is strongly convex in $\mathbf{h}$ for $\mu>0$, and also Lipschitz for bounded edge flows, filter coefficient $\mathbf{h}$ and the matrix $[\mathbf{F}_t]_{e_t,:}$, for all t. The gradient of $\ell_t(\mathbf{h})$ w.r.t. $\mathbf{h}$ can be found as

$$\nabla_{\mathbf{h}}\ell_t(\mathbf{h})=[\mathbf{F}_t]_{e_t,:}^\top[\mathbf{F}_t]_{e_t,:}\mathbf{h}-[\mathbf{F}_t]_{e_t,:}^\top[\mathbf{f}_t]_{e_t}+\mu\mathbf{h}. \tag{10}$$

Then, we can update the filter coefficient $\mathbf{h}_t$ from the previously learned $\mathbf{h}_{t-1}$ via an online gradient descent step

$$\begin{aligned}\mathbf{h}_t&=\mathbf{h}_{t-1}-\eta\nabla_{\mathbf{h}}\ell_t(\mathbf{h}_{t-1})\\&=((1-\mu\eta)\mathbf{I}-\eta_t[\mathbf{F}_t]_{e_t,:}^\top[\mathbf{F}_t]_{e_t,:})\mathbf{h}_{t-1}+\eta[\mathbf{F}_t]_{e_t,:}^\top[\mathbf{f}_t]_{e_t},\end{aligned} \tag{11}$$

with a learning rate $\eta>0$.

Note that the filter coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ control the lower and upper adjacencies in the prediction independently. We can equivalently write the loss function (9) as

$$\ell_t(\mathbf{h})=([\mathbf{F}_{\mathrm{d},t}]_{e_t,:}\boldsymbol{\alpha}+[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}\boldsymbol{\beta}-[\mathbf{f}_t]_{e_t})^2+\mu(\|\boldsymbol{\alpha}\|_2^2+\|\boldsymbol{\beta}\|_2^2). \tag{12}$$

The gradients of $\ell_t(\mathbf{h})$ w.r.t $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be found as

$$\begin{aligned}\nabla_{\boldsymbol{\alpha}}\ell_t(\mathbf{h})&=[\mathbf{F}_{\mathrm{d},t}]_{e_t,:}^\top[\mathbf{F}_{\mathrm{d},t}]_{e_t,:}\boldsymbol{\alpha}-[\mathbf{F}_{\mathrm{d},t}]_{e_t,:}^\top[\mathbf{f}_t]_{e_t}\\&\quad+\mu\boldsymbol{\alpha}+[\mathbf{F}_{\mathrm{d},t}]_{e_t,:}^\top[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}\boldsymbol{\beta},\\\nabla_{\boldsymbol{\beta}}\ell_t(\mathbf{h})&=[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}^\top[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}\boldsymbol{\beta}-[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}^\top[\mathbf{f}_t]_{e_t}\\&\quad+\mu\boldsymbol{\beta}+[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}^\top[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}\boldsymbol{\alpha},\end{aligned} \tag{13}$$

where $[\mathbf{F}_{\mathrm{d},t}]_{e_t,:}^\top[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}=[\bar{\mathbf{f}}_t]_{e_t}^2$ since the lower shifted signal $\bar{\mathbf{f}}_{\mathrm{d},t}^{(k)}$ is independent to the upper shifted one $\bar{\mathbf{f}}_{\mathrm{u},t}^{(k)}$ based on the fact that $\mathbf{B}_1\mathbf{B}_2=\mathbf{0}$ [10]. When the default value on edge e_t is 0, i.e., $[\bar{\mathbf{f}}_t]_{e_t}=g=0$, then term $[\mathbf{F}_{\mathrm{d},t}]_{e_t,:}^\top[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}\boldsymbol{\beta}_{t-1}$ disappears in $\nabla_{\boldsymbol{\alpha}}\ell_t(\mathbf{h})$ and likewise $[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}^\top[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}\boldsymbol{\alpha}$ disappears in $\nabla_{\boldsymbol{\beta}}\ell_t(\mathbf{h})$, making the updates of $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ independent of each other. Then, the online update of the two sets of coefficients is given by

$$\begin{aligned}\boldsymbol{\alpha}_t&=\boldsymbol{\alpha}_{t-1}-\eta_{\mathrm{d}}\nabla_{\boldsymbol{\alpha}}\ell_t(\mathbf{h}_{t-1}),\\\boldsymbol{\beta}_t&=\boldsymbol{\beta}_{t-1}-\eta_{\mathrm{u}}\nabla_{\boldsymbol{\beta}}\ell_t(\mathbf{h}_{t-1}),\end{aligned} \tag{14}$$

where we consider learning rates η_{d} and η_{u} for the lower and upper filter coefficients. When computing $\nabla_{\boldsymbol{\alpha}}\ell_t(\mathbf{h}_{t-1})$ in (14), the major computational cost comes from the terms $[\mathbf{F}_{\mathrm{d},t}]_{e_t,:}^\top[\mathbf{F}_{\mathrm{d},t}]_{e_t,:}\boldsymbol{\alpha}$ and $[\mathbf{F}_{\mathrm{d},t}]_{e_t,:}^\top[\mathbf{F}_{\mathrm{u},t}]_{e_t,:}\boldsymbol{\beta}_{t-1}$, which are of orders $\mathcal{O}(2K_{\mathrm{d}})$ and $\mathcal{O}(K_{\mathrm{u}}+K_{\mathrm{d}})$, respectively, and likewise for computing $\nabla_{\boldsymbol{\beta}}\ell_t(\mathbf{h}_{t-1})$.

Compared to (11), the updates in (14) uncouple the filter learning into the lower and upper simplicial adjacencies. In the simplicial frequency domain, as studied in [10], this implies that we update the filter frequency response in two orthogonal edge flow subspaces given by the Hodge decomposition, namely the gradient and the curl space. The update of the filter coefficients $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ concludes the online filter learning at time t, which is followed by the new prediction $[\hat{\mathbf{f}}_{t+1}]_{e_{t+1}}=[\mathbf{F}_{t+1}\mathbf{h}_t]_{e_{t+1}}$ for next time instance $t+1$.

Regret analysis. To analyze the performance of the online filter learning [cf. (11)] w.r.t. the offline batch solution [cf. (8)] optimized upon observing the entire edge flow sequence, we perform a regret analysis. The static regret of an online learning is defined as [22, 23]

$$R_T(\mathbf{h}):=\sum_{t=0}^{T}l_t(\mathbf{h}_t)-\min_{\mathbf{h}}\sum_{t=0}^{T}l_t(\mathbf{h}) \tag{15}$$

where the first term is the accumulated loss of the online learner [cf. (11)] with the sequence of filters $\{\mathbf{h}_0,\dots,\mathbf{h}_T\}$ and the second is the minimum of the loss upon observing the whole sequence, which can be obtained by the batch solution $\mathbf{h}^*$ at $t=T$ [cf. (8)]. The following proposition gives an upper bound for $R_T(\mathbf{h})$ based on the convexity of the loss in (9) and the Lipschitz property $\|\nabla_{\mathbf{h}}\ell_t(\mathbf{h})\|_2\leq L$. This upper bound indicates the worst-case performance gap of the online filter learning relative to an optimal $\mathbf{h}^*$.

Proposition 1. *Given a sequence of coming edge flows* $\mathbf{f}_t$, *for* $t=0,\dots,T$, *consider a sequence of online filters* $\{\mathbf{h}_t\}$ *updated over*

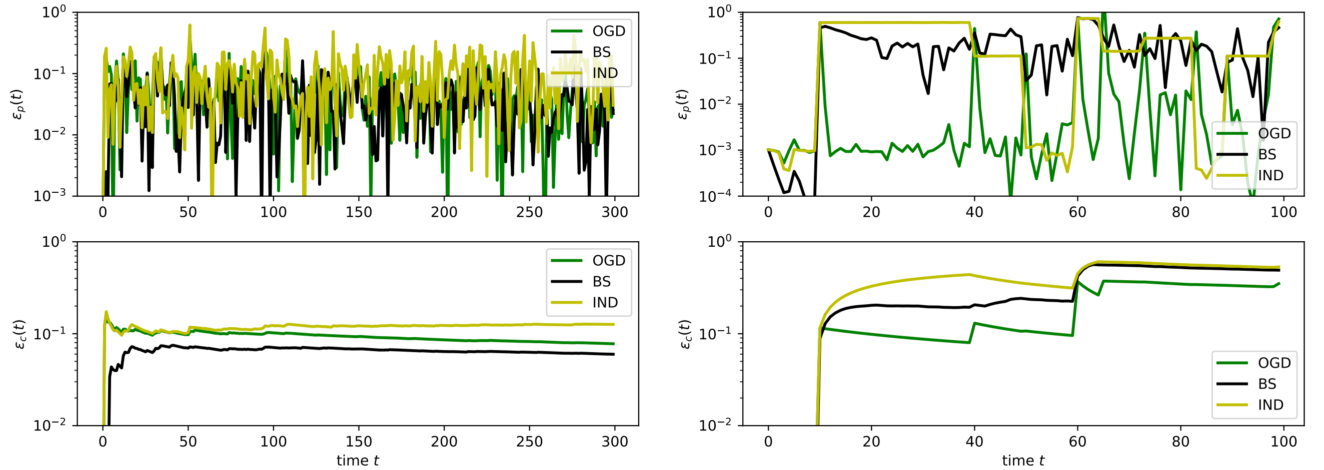

Fig. 2. Prediction performance of the online gradient descent (OGD), the batch solution (BS) and the inductive filter (IND) on an expanding synthetic SC (*(left)*) and an expanding coauthorship complex (*(right)*) in terms of $\epsilon_p(t)$ *(top)* and $\epsilon_c(t)$ *(bottom)*.

a sequence of L Lipschitz functions $l_t(\mathbf{h}_t)$ [cf. (9)] with a fixed learning rate η. The static regret of the online filter learning w.r.t an optimal simplicial filter $\mathbf{h}^\star$ is upper bounded as

$$R_T(\mathbf{h}^\star) \leq \frac{\|\mathbf{h}^\star\|_2^2}{2\eta} + \frac{\eta}{2} L^2 T. \tag{16}$$

The proof can be derived from [22, Thm. 2.13]. This bound obtains the minimum $R_T^*(\mathbf{h}^*) = \|\mathbf{h}^*\|_2 L\sqrt{T}$, for an optimal $\eta^* = \|\mathbf{h}^*\|_2/L\sqrt{T}$, which grows sub-linearly with T. This implies the difference between the average loss of the online learner and the average loss of the optimal $\mathbf{h}^\star$ tends to zero as T goes to infinity.

4. NUMERICAL EXPERIMENTS

We corroborate the online edge flow predictor on expanding SCs on a synthetic dataset and a coauthorship dataset. As a baseline, we considered the offline batch solution (BS) [cf. (8)] optimized based on a least-squares solution. Both datasets contain nodes, edges, and triangles, as well as edge flows. We remove the last T edges based on their index and the respective triangles which they participate in. The remaining SC and the edge flow form the initial state. The SC expands with one edge e_t coming at time t, and the edge flow on e_t is assumed to be the median of the known edge flows $\mathbf{f}_{t-1}$. To enable a warm start of the simplicial filter, we use the first coming edge flow to pretrain the filter coefficient $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ by solving (8), which are also used to build an inductive filter (IND) as another baseline, i.e., $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ not updated over time.

We evaluate the performance by using a pointwise and a cumulative root normalized mean square error, ϵ_p and ϵ_c, respectively, which measure the instantaneous and overall errors, given by

$$\epsilon_p(t) = \frac{|[\hat{\mathbf{f}}_t]_{e_t} - [\mathbf{f}_t]_{e_t}|}{[\mathbf{f}_t]_{e_t}},\ \epsilon_c(t) = \sqrt{\frac{\sum_{\tau=0}^{t}([\hat{\mathbf{f}}_\tau]_{e_\tau} - [\mathbf{f}_\tau]_{e_\tau})^2}{\sum_{\tau=0}^{t}[\hat{\mathbf{f}}_\tau]_{e_\tau}^2}}.$$

Synthetic dataset. We generated an alpha complex (AC) of 300 nodes, 850 edges and 551 triangles with a filtration of 0.01 using the Gudhi toolbox [27, 28]. The edge flow on the AC is generated to have a low-pass nature, as $\mathbf{f} = (\mathbf{I} + 0.5\mathbf{L}_\mathrm{d} + 0.3\mathbf{L}_\mathrm{u})^{-1}\mathbf{f}_0$ where $\mathbf{f}_0 = 5\mathbf{1}$. After removing the last 300 edges, the AC contains 550 edges and 194 triangles. At each time instance $t = 0, 1, \ldots, 299$, we randomly added one edge to the AC and update $\mathbf{B}_{1,t}$ and $\mathbf{B}_{2,t}$. For the prediction, we set the filter orders $K_\mathrm{d} = K_\mathrm{u} = 5$ where the lower and upper Laplacians are normalized by the maximal eigenvalue of $\mathbf{L}_1$. We set $\mu = 0.1$ and $\eta_{\mathrm{d},t} = \eta_{\mathrm{u},t} = 0.01$.

As reported in Fig. 3 (left), the performance of the OGD sits in between the BS and the IND and it approaches to the BS while becoming better than the IND over time in terms of both $\epsilon_p(t)$ and $\epsilon_c(t)$. This learning ability of the OGD leads to an overall error $\epsilon_c(t) = 0.09$ compared to 0.12 for the IND and 0.07 for the BS at the final time instance $t = 299$.

Coauthorship prediction. We then considered a coauthorship complex from the Semantic Scholar Open Research Corpus [29] where nodes are authors and papers with k-authors are represented as k-simplices. As processed in [30], there are 352 nodes, 1472 edges and 3285 triangles. We aim to predict the edge flows which are the citation numbers of the two-author papers. First, we removed the last 100 edges, resulting in 1374 edges and 2722 triangles and then added one edge each time based on their index in an ascending order. Other parameters in the OGD remain the same as in the synthetic case.

From Fig. 3 (right), we observe that the OGD performs better than both the BS and the IND in general in terms of both $\epsilon_p(t)$ and $\epsilon_c(t)$. At some time instances around $t = 10, 40$ and 60, the pointwise errors $\epsilon_p(t)$ of all three approaches present drastic increases, which leads to the increase in $\epsilon_c(t)$ as well. This is due to the piecewise constant nature of the underlying edge flow with large discontinuity at these time instances [29]. While optimizing the predictor over the entire available sequence, the BS performance is deviated by the large discontinuity. However, when learning based on the instantaneous error of the incoming edge flow, the OGD is not influenced by the large discontinuity at other time instances when edge flows respect the slow-varying property, e.g., between $t = 10$ and 40. This leads to a 90% prediction accuracy of the OGD compared to 30% and 26% for the BS and the IND, where a prediction is considered accurate if it is within $\pm 10\%$ of the true value [30].

5. CONCLUSION

In this paper, we proposed an online edge flow prediction algorithm for expanding simplicial complexes with edges and edge flows coming over time. The algorithm leverages the simplicial filter to predict the incoming edge flow on the updated topology. This prediction admits a distributed implementation with a low communication cost. When an edge flow is revealed, we then apply an online gradient descent to learn the filter coefficient, which is shown to approach the batch solution asymptotically. This online edge flow prediction can be easily generalized to other higher-order simplicial signal spaces. Future works can be focused to analyze the predictor from the spectral perspective or to study the online prediction when the existing edge flows also vary over time.

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 Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

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

[3] W. Huang, T. A. Bolton, J. D. Medaglia, D. S. Bassett, A. Ribeiro, and D. Van De Ville, "A graph signal processing perspective on functional brain imaging," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 868–885, 2018.

[4] K. K. Leung, W. A. Massey, and W. Whitt, "Traffic models for wireless communication networks," *IEEE Journal on selected areas in Communications*, vol. 12, no. 8, pp. 1353–1364, 1994.

[5] M. T. Schaub and S. Segarra, "Flow smoothing and denoising: Graph signal processing in the edge-space," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 735–739.

[6] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye, "Statistical ranking and combinatorial hodge theory," *Mathematical Programming*, vol. 127, no. 1, pp. 203–244, 2011.

[7] O. Candogan, I. Menache, A. Ozdaglar, and P. A. Parrilo, "Flows and decompositions of games: Harmonic and potential games," *Mathematics of Operations Research*, vol. 36, no. 3, pp. 474–503, 2011.

[8] T. Gebhart, X. Fu, and R. J. Funk, "Go with the flow? a large-scale analysis of health care delivery networks in the united states using hodge theory," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 3812–3823.

[9] A. Mock and I. Volić, "Political structures and the topology of simplicial complexes," *Mathematical Social Sciences*, vol. 114, pp. 39–57, 2021.

[10] M. Yang, E. Isufi, M. T. Schaub, and G. Leus, "Simplicial convolutional filters," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4633–4648, 2022.

[11] L.-H. Lim, "Hodge laplacians on graphs," *SIAM Review*, vol. 62, no. 3, pp. 685–715, 2020.

[12] M. T. Schaub, Y. Zhu, J.-B. Seby, T. M. Roddenberry, and S. Segarra, "Signal processing on higher-order networks: Livin'on the edge... and beyond," *Signal Processing*, vol. 187, pp. 108149, 2021.

[13] S. Barbarossa and S. Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2992–3007, 2020.

[14] M. Yang and E. Isufi, "Simplicial trend filtering," in *IEEE Asilomar Conference on Signals, Systems and Computers*. 2022, submitted for publication.

[15] O. T. Courtney and G. Bianconi, "Weighted growing simplicial complexes," *Physical Review E*, vol. 95, no. 6, pp. 062301, 2017.

[16] K. Kovalenko, I. Sendiña-Nadal, N. Khalil, A. Dainiak, D. Musatov, A. M. Raigorodskii, K. Alfaro-Bittner, B. Barzel, and S. Boccaletti, "Growing scale-free simplices," *Communications Physics*, vol. 4, no. 1, pp. 1–9, 2021.

[17] S. Oh, Y. Lee, J. Lee, and B. Kahng, "Emergence of betti numbers in growing simplicial complexes: analytical solutions," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2021, no. 8, pp. 083218, 2021.

[18] K. Sugishita, N. Abdel-Mottaleb, Q. Zhang, and N. Masuda, "A growth model for water distribution networks with loops," *Proceedings of the Royal Society A*, vol. 477, no. 2255, pp. 20210528, 2021.

[19] M. Yang, E. Isufi, and G. Leus, "Simplicial convolutional neural networks," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8847–8851.

[20] J. Jia, M. T. Schaub, S. Segarra, and A. R. Benson, "Graph-based semi-supervised & active learning for edge flows," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 761–771.

[21] M. Yang, E. Isufi, M. T. Schaub, and G. Leus, "Finite Impulse Response Filters for Simplicial Complexes," in *2021 29th European Signal Processing Conference (EUSIPCO)*, Aug. 2021, pp. 2005–2009, ISSN: 2076-1465.

[22] F. Orabona, "A modern introduction to online learning," *arXiv preprint arXiv:1912.13213*, 2019.

[23] S. Shalev-Shwartz et al., "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[24] Y. Shen, G. Leus, and G. B. Giannakis, "Online Graph-Adaptive Learning With Scalability and Privacy," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2471–2483, May 2019.

[25] B. Das and E. Isufi, "Online filtering over expanding graphs," in *2022 56th Asilomar Conference on Signals, Systems, and Computers*. 2022, submitted for publication.

[26] B. Das and E. Isufi, "Graph filtering over expanding graphs," in *2022 IEEE Data Science and Learning Workshop (DSLW)*, 2022, pp. 1–8.

[27] The GUDHI Project, *GUDHI User and Reference Manual*, GUDHI Editorial Board, 3.6.0 edition, 2022.

[28] V. Rouvreau, "Alpha complex," in *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.6.0 edition, 2022.

[29] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, et al., "Construction of the literature graph in semantic scholar," *arXiv preprint arXiv:1805.02262*, 2018.

[30] S. Ebli, M. Defferrard, and G. Spreemann, "Simplicial neural networks," in *TDA & Beyond*, 2020.