

Simplicial Vector Autoregressive Model For Streaming Edge Flows

Krishnan, Joshin; Money, Rohan; Beferull-Lozano, Baltasar; Isufi, Elvin

DOI

[10.1109/ICASSP49357.2023.10096095](https://doi.org/10.1109/ICASSP49357.2023.10096095)

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)

Citation (APA)

Krishnan, J., Money, R., Beferull-Lozano, B., & Isufi, E. (2023). Simplicial Vector Autoregressive Model For Streaming Edge Flows. In *Proceedings of the ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings; Vol. 2023-June). IEEE.
<https://doi.org/10.1109/ICASSP49357.2023.10096095>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

SIMPLICIAL VECTOR AUTOREGRESSIVE MODEL FOR STREAMING EDGE FLOWS

Joshin Krishnan*, Rohan Money†, Baltasar Beferull-Lozano*, and Elvin Isufi‡.

*SIGIPRO Department, Simula Metropolitan Center for Digital Engineering, Oslo, Norway,

†WISENET Center, Department of ICT, University of Agder, Grimstad, Norway,

‡ Intelligent Systems Department, Delft University of Technology, Delft, The Netherlands,

joshin@simula.no; rohamtm@uia.no; baltasar.beferull@uia.no; baltasar@simula.no; e.isufi-1@tudelft.nl.

ABSTRACT

Vector autoregressive (VAR) model is widely used to model time-varying processes, but it suffers from prohibitive growth of the parameters when the number of time series exceeds a few hundreds. We propose a simplicial VAR model to mitigate the curse of dimensionality of the VAR models when the time series are defined over higher-order network structures such as edges, triangles, etc. The proposed model shares parameters across the simplicial signals by leveraging the simplicial convolutional filter and captures structure-aware spatio-temporal dependencies of the time-varying processes. Targetting the streaming signals from the real-world nonstationary networks, we develop a group-lasso-based online strategy to learn the proposed model. Using traffic and water distribution networks, we demonstrate that the proposed model achieves competitive signal prediction accuracy with a significantly less number of parameters than the VAR models.

Index Terms— Simplicial convolution, simplicial vector autoregressive model, Hodge Laplacians.

1. INTRODUCTION

An ever-increasing number of applications involving social, biological, financial, industrial, water, power, communication, and transportation networks call for modelling the time-varying processes generated by them [1–3]. One of the most successful and tractable approaches is the vector autoregressive model (VAR) and variations therein [4–10], which relies on the past realizations of the process, and determine the future values as a linear combination of such spatio-temporal realizations.

Despite their wide applicability, vanilla VAR models ignore any underlying structure in the data. This leads to a high number of VAR parameters and unaffordable computational complexity, when the number of time series exceeds a few tens or hundreds. However, the graph-based VAR (G-VAR) [11] models mitigate this curse of dimensionality by considering the time series as processes over the vertices of a graph and modelling their evolution as a sparse linear combination of the signals in the adjacent vertices. G-VAR leverages the graph and graph convolutional filters as effective inductive biases to reduce the degrees of freedom, resulting in structure-aware sparsity and a low number of parameters.

While the G-VAR alleviates the challenges of the vanilla VAR models for vertex time series, network data are commonly present also on different higher-order network structures, such as edges, triangles, and so on [12, 13]. For instance, the flows in a water network

can be intuitively defined as a process evolving over the edges of a network rather than on its vertices. A recent body of literature has developed structure-aware data processing techniques by leveraging simplicial complexes [12, 14–17]. A simplicial complex is a mathematical representation of the higher-order connectivities of the networks, and allows for mathematical tractability in a way akin to graphs for pairwise similarities [18]. Although, the popular signal processing concepts such as the Fourier transform, sampling theory [19], and convolution filters [20] are developed recently for the simplexes, they can be seen as solutions to process *time-invariant* data, and ignore the spatio-temporal coupling present in simplicial processes.

This paper proposes a structure-aware VAR model for *time-varying* processes defined over simplices and a data-driven strategy to learn the model. One major challenge in learning the model is that a batch-based offline strategy requires processing the entire batch of data, which is not suitable since real-world networks are often non-stationary and the data is available only in a streaming manner [21]. Hence, we propose an online strategy for learning the VAR model over the simplices. Our specific contribution is threefold:

1. We propose a simplicial VAR (S-VAR) model for time-varying processes to capture sparse spatio-temporal dependencies between the simplicial signals.
2. The S-VAR eliminates the curse of dimensionality by leveraging simplicial convolution filters as inductive biases, attributed with locality and parameter sharing between simplices. This allows the proposed methods to be implemented efficiently with a cost that is linear in the number of simplices and to have an order-one number of parameters.
3. We propose an online method for learning the S-VAR models using a group-lasso-based optimization framework, solved via composite objective mirror descent.

2. PRELIMINARIES

2.1. Vector Autoregressive Models

A VAR model represents the evolution of a multivariate time-varying process $\mathbf{x}_t \in \mathbb{R}^N$ as a linear combination of its P past realizations $\mathbf{x}_{t-1}, \dots, \mathbf{x}_{t-P}$, i.e.,

$$\mathbf{x}_t = \sum_{p=1}^P \mathbf{W}_p \mathbf{x}_{t-p} + \boldsymbol{\varepsilon}_t, \quad (1)$$

where $\mathbf{W}_p \in \mathbb{R}^{N \times N}$ collects the parameters for the temporal lag p , in which $[\mathbf{W}_p]_{ij}$ captures the spatio-temporal dependency between $[\mathbf{x}_{t-p}]_j$ and $[\mathbf{x}_t]_i$. The variable $\boldsymbol{\varepsilon}_t$ is the model uncertainty, and is typically considered zero-mean Gaussian.

The study was supported by the IKTPLUSS INDURB grant 270730/O70 and the TU Delft AI Labs programme.

Model (1) has two major limitations: *i*) it involves a total of N^2P parameters, which becomes prohibitively high in real-world networks, where the values of N are in the order of a few hundreds or more, and *ii*) it ignores any underlying network structure; hence, if a node (or edge) disappears/appears, it has to be retrained. To mitigate this issue, we propose a simplex-based VAR model with reduced degrees of freedom by exploiting the simplicial structure as an inductive bias to not pay in representation power.

2.2. Simplicial Complex and Simplicial Signals

Let $\mathcal{V} = \{1, \dots, N\}$ be a set of vertices. A k -simplex $\mathcal{S}^k$ is a subset of $\mathcal{V}$ containing $k+1$ distinct elements. A simplicial complex (SC) of order K , denoted as $\mathcal{X}^K$, is a set of simplices such that at least one K -simplex is an element of $\mathcal{X}^K$, and a simplex $\mathcal{S}^k$ is an element of $\mathcal{X}^K$, if and only if all the subsets of $\mathcal{S}^k$ are also the elements of $\mathcal{X}^K$ [18]. An example for $\mathcal{X}^2$ is provided in Fig. 1.

Adjacencies in a SC can be encapsulated using incidence matrices and Hodge Laplacians [18]. Let N_k be the number of k -simplices in a SC. The incidence matrix $\mathbf{B}_k \in \mathbb{R}^{N_{k-1} \times N_k}$ has $(k-1)$ -simplices as the row index, and k -simplices as the column index, capturing the adjacencies between them. For instance, $\mathbf{B}_1$ is the node-to-edge incidence matrix and $\mathbf{B}_2$ is the edge-to-triangle incidence matrix. These incidence matrices satisfy the so-called boundary condition $\mathbf{B}_1\mathbf{B}_2 = \mathbf{0}$. The Hodge Laplacians representing the structure of $\mathcal{X}^2$ are given by

$$\begin{aligned} \mathbf{L}_0 &= \mathbf{B}_1\mathbf{B}_1^\top, \\ \mathbf{L}_1 &= \mathbf{L}_{1,\ell} + \mathbf{L}_{1,u} := \mathbf{B}_1^\top\mathbf{B}_1 + \mathbf{B}_2\mathbf{B}_2^\top, \\ \mathbf{L}_2 &= \mathbf{B}_2^\top\mathbf{B}_2. \end{aligned} \quad (2)$$

Here, $\mathbf{L}_0$ is the popular graph Laplacian [22] that expresses adjacencies between vertices based on common edges. The Hodge Laplacian $\mathbf{L}_1$ expresses adjacencies between edges based on *i*) the common vertices via the lower-Laplacian $\mathbf{L}_{1,\ell} = \mathbf{B}_1^\top\mathbf{B}_1$, and *ii*) the triangles via the upper-Laplacian $\mathbf{L}_{1,u} = \mathbf{B}_2\mathbf{B}_2^\top$; and $\mathbf{L}_2$ expresses proximities between triangles via lower edge adjacencies.

Simplicial signals are defined as functions from any k -simplices to the set of real numbers. For instance, over the edges, we have an edge flow $\mathbf{f} = [f_1, \dots, f_{N_1}]^\top \in \mathbb{R}^{N_1}$ with entry f_e being the flow of the edge $e = (m, n)$ in $\mathcal{S}^1$. Similarly, we define the signals over the vertex as $\mathbf{v} \in \mathbb{R}^{N_0}$ and the triangles as $\boldsymbol{\tau} \in \mathbb{R}^{N_2}$; see Fig. 1. The proximities in a SC are translated into proximities between the signals defined over it, and the objective therefore is to exploit such proximities to process signals [19].

2.3. Convolution in the Simplex

We can process a simplicial signal with simplicial convolutional filters, which for edge signals has the form [20, 23]

$$\mathbf{f}_0 = \underbrace{\sum_{k=0}^{K_\ell} \beta_k^\ell \mathbf{L}_{1,\ell}^k \mathbf{f}}_{\mathbf{A}(\mathbf{L}_{1,\ell})} + \underbrace{\sum_{k=1}^{K_u} \beta_k^u \mathbf{L}_{1,u}^k \mathbf{f}}_{\mathbf{B}(\mathbf{L}_{1,u})}, \quad (3)$$

where $\{\beta_k^\ell\}_{k=1}^{K_\ell}$ and $\{\beta_k^u\}_{k=1}^{K_u}$ are respectively the filter coefficients weighting $\mathbf{f}$ shifted by $\mathbf{L}_{1,\ell}$ up to K_ℓ hops and $\mathbf{L}_{1,u}$ up to K_u hops; see Fig. 2. Here, β_0^ℓ is the filter coefficient weighting the original signal $\mathbf{f}$, and is grouped with $\{\beta_k^\ell\}_{k=1}^{K_\ell}$ for convenience of representation. The total number of parameters involved in (3) is $K := K_\ell + K_u + 1$. By defining *simplicial convolutional filtering* matrix $\mathbf{H}(\mathbf{L}_1) := \mathbf{A}(\mathbf{L}_{1,\ell}) + \mathbf{B}(\mathbf{L}_{1,u})$, we can compactly write (3) as

$$\mathbf{f}_0 = \mathbf{H}(\mathbf{L}_1)\mathbf{f}. \quad (4)$$

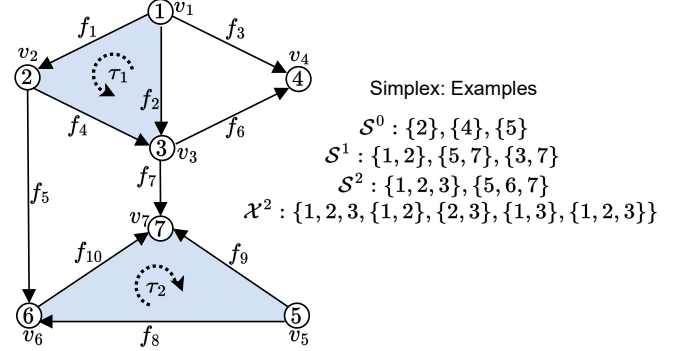


Fig. 1: A geometrical representation of a simplicial complex and associated simplicial signals.

This convolutional filter shares the parameters across the different edges, irrespective of their labelling and flow orientation. Although we focus on the edge signals, all the methods developed in the sequel can be easily extended to any simplicial level by using a filter $\mathbf{H}(\mathbf{L}_k)$ in a form similar to (4).

3. SIMPLICIAL VECTOR AUTOREGRESSIVE MODEL

We define an S-VAR model of order P for time-varying edge flow processes $\mathbf{f}_t$:

$$\mathbf{f}_t = \sum_{p=1}^P \mathbf{H}_p(\mathbf{L}_1)\mathbf{f}_{t-p} + \boldsymbol{\varepsilon}_t, \quad (5)$$

where $\{\mathbf{H}_p(\mathbf{L}_1)\}_p$ are simplicial convolutional filters of the form (3) defined by parameters $\{\beta_{p,k}^\ell\}_{k=0}^{K_\ell}$ and $\{\beta_{p,k}^u\}_{k=1}^{K_u}$. Here, the filter $\mathbf{H}_p(\mathbf{L}_1)$ captures the spatial dependencies of the process $\mathbf{f}_{t-p}$, and the VAR regression captures the temporal dependencies up to a time lag P . Model (5) uses the simplicial structure as the inductive bias by leveraging the simplicial convolution filters. The inductive bias enables the model to take advantage of the structure-aware proximities of the network, resulting in a low number of parameters. The S-VAR model involves only KP parameters, irrespective of the number of the process N . This gives a clear advantage to S-VAR in real-world networks having a few hundreds or more sensors, compared to the vanilla VAR (1) with N^2P parameters.

Let $\boldsymbol{\beta}_p := [\beta_{p,0}^\ell, \dots, \beta_{p,K_\ell}^\ell, \beta_{p,1}^u, \dots, \beta_{p,K_u}^u] \in \mathbb{R}^K$, and $\tilde{\mathbf{F}}_{t-p} := [\mathbf{L}_{1,\ell}^0 \mathbf{f}_{t-p}, \mathbf{L}_{1,\ell}^1 \mathbf{f}_{t-p}, \dots, \mathbf{L}_{1,\ell}^{K_\ell} \mathbf{f}_{t-p}, \mathbf{L}_{1,u}^1 \mathbf{f}_{t-p}, \dots, \mathbf{L}_{1,u}^{K_u} \mathbf{f}_{t-p}]$, where, the matrix $\tilde{\mathbf{F}}_{t-p} \in \mathbb{R}^{N_1 \times K}$ collects the shifted versions of the edge signal $\mathbf{f}_{t-p}$. Define $\mathbf{F}_t := [\tilde{\mathbf{F}}_{t-1}, \dots, \tilde{\mathbf{F}}_{t-P}] \in \mathbb{R}^{N_1 \times KP}$ and $\boldsymbol{\beta} := [\boldsymbol{\beta}_1^\top, \dots, \boldsymbol{\beta}_P^\top]^\top \in \mathbb{R}^{KP}$, using which, the S-VAR model (5) can be compactly written as

$$\mathbf{f}_t = \mathbf{F}_t\boldsymbol{\beta} + \boldsymbol{\varepsilon}_t. \quad (6)$$

3.1. Batch Estimation

The S-VAR parameters can be estimated by minimizing the empirical mean square error (MSE) over T_0 samples as

$$\text{MSE} = \frac{1}{2} \sum_{t=P+1}^{T_0} \|\mathbf{f}_t - \mathbf{F}_t\boldsymbol{\beta}\|_2^2 + \Omega(\boldsymbol{\beta}, \mathbf{L}_1), \quad (7)$$

where $\Omega(\boldsymbol{\beta}, \mathbf{L}_1)$ is a regularizer to avoid over fitting. The regularizer consists of three terms: *i*) $\hat{\mathbf{f}}_t \mathbf{L}_{1,\ell} \hat{\mathbf{f}}_t$, where $\hat{\mathbf{f}}_t = \mathbf{F}_t\boldsymbol{\beta}$, *ii*) $\hat{\mathbf{f}}_t \mathbf{L}_{1,u} \hat{\mathbf{f}}_t$, and *iii*) $\sum_{p=1}^P \|\boldsymbol{\beta}_p\|_2$. Here, *i*) and *ii*) impose constraints based on the simplicial structure, e.g., $\hat{\mathbf{f}}_t \mathbf{L}_1 \hat{\mathbf{f}}_t = \|\mathbf{B}_1 \hat{\mathbf{f}}_t\|^2$ regularizes the divergence flows ($\mathbf{B}_1 \hat{\mathbf{f}}_t$), thereby enforcing the conservation of the

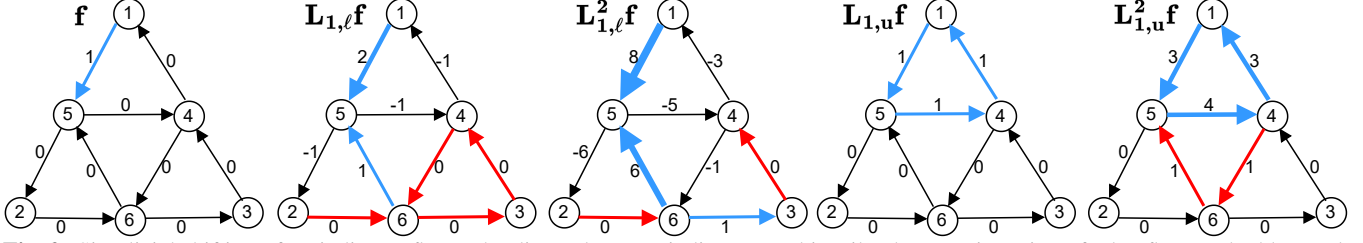


Fig. 2: Simplicial shifting of an indicator flow. The directed arrows indicate an arbitrarily chosen orientation of edge flows. The blue, red, and black colours respectively indicate high, intermediate, and low values of flows, which are further distinguished by edge thicknesses.

flows at the nodes, and $\hat{\mathbf{f}}_t \mathbf{L}_{1,u} \hat{\mathbf{f}}_t = \|\mathbf{B}_2^\top \hat{\mathbf{f}}_t\|^2$ regularizes the cyclic flows ($\mathbf{B}_2^\top \hat{\mathbf{f}}_t$) [12, 20]. The third one is a group-lasso regularizer, acting along the temporal domain by imposing group sparsity on $\{\beta_p\}_{p=1}^P$. The S-VAR parameters can be learned using a regularized optimization,

$$\hat{\beta} = \arg \min_{\beta} \sum_{t=P+1}^{T_0} h_t(\beta) + \lambda \sum_{p=1}^P \|\beta_p\|_2, \quad (8)$$

where

$$h_t(\beta) = \frac{1}{2} \left[\|\mathbf{f}_t - \mathbf{F}_t \beta\|_2^2 + (\mathbf{F}_t \beta)^\top (\mu_1 \mathbf{L}_{1,\ell} + \mu_2 \mathbf{L}_{1,u}) (\mathbf{F}_t \beta) \right], \quad (9)$$

and $\mu_1, \mu_2, \lambda \geq 0$ are the hyperparameters associated with the regularizers. Here, $h_t(\cdot)$ is a regularized loss function, but hereafter for brevity, we use the term *loss function*.

3.2. Online Estimation

Note that (8) is an offline (batch) strategy, meaning that the entire batch of data $\mathbf{f}_t$, $t = 1, \dots, T_0$ is required to compute the solution. When the flows are time varying and appear in an online fashion, we are interested in an online update of the parameters as data appear; and this will also tackle the computational challenges of the batch estimation. Next, we develop an online strategy to learn the S-VAR.

We replace the cumulative loss $\sum_{t=P+1}^{T_0} h_t(\beta)$ with a running average loss using an exponential window [4]:

$$\ell_t(\beta) = \delta \sum_{t'=P+1}^t \gamma^{t-t'} h_{t'}(\beta), \quad (10)$$

where $\gamma \in (0, 1)$ is the forgetting factor of the window, and $\delta = 1 - \gamma$ is set to normalize the exponential weighting window. Note that (10) resembles the typical loss function in a recursive least square (RLS) problem. First, we expand (10) using (9) as

$$\ell_t(\beta) = \frac{1}{2} \delta \sum_{t'=P}^{t-1} \gamma^{t-t'} \|\mathbf{f}_{t'}\|_2^2 + \frac{1}{2} \beta^\top \Phi_t \beta - \mathbf{r}_t^\top \beta, \quad (11)$$

where

$$\Phi_t = \delta \sum_{t'=P+1}^t \gamma^{t-t'} \mathbf{F}_{t'}^\top (\mathbf{I} + \mu_1 \mathbf{L}_{1,\ell} + \mu_2 \mathbf{L}_{1,u}) \mathbf{F}_{t'}, \quad (12)$$

$$\mathbf{r}_t = \delta \sum_{t'=P+1}^t \gamma^{t-t'} \mathbf{F}_{t'}^\top \mathbf{f}_{t'}. \quad (13)$$

The batch estimate of the S-VAR parameters using the exponential window up to time t can be found by

$$\hat{\beta}_t = \arg \min_{\beta} \ell_t(\beta) + \omega(\beta), \quad (14)$$

where $\omega(\beta) = \lambda \sum_{p=1}^P \|\beta_p\|_2$. It is straightforward to verify that $\ell_t(\cdot)$ and $\omega(\cdot)$ are convex functions, however, note that $\ell_t(\cdot)$ is a differentiable function, whereas $\omega(\cdot)$ is not differentiable. The online subgradient descent (OSGD) or the mirror descent (MD) methods can be used to solve (14) online. However, these methods work by linearizing $\ell_t(\cdot) + \omega(\cdot)$ using a subgradient $g \in \partial(\ell_t(\cdot) + \omega(\cdot))$.

A linearized $\omega(\cdot)$ loses its ability to induce sparsity, resulting in non-sparse estimates. Hence, we choose an alternate optimization technique known as composite objective mirror descent (COMID) [4–7, 24], a modified version of the MD algorithm, in which the differentiable part $\ell_t(\cdot)$ is linearized, whereas $\omega(\cdot)$ is kept intact. The online COMID update is given by

$$\hat{\beta}[t+1] = \arg \min_{\beta} \nabla \ell_t(\hat{\beta}[t])^\top \beta + \frac{1}{2\eta_t} B(\hat{\beta}[t], \beta) + \omega(\beta), \quad (15)$$

where $\hat{\beta}[t]$ denotes the estimate of β obtained at time t , $\nabla \ell_t(\hat{\beta}[t])$ is the gradient of $\ell_t(\cdot)$ at $\hat{\beta}[t]$, computed from (11) as

$$\nabla \ell_t(\hat{\beta}[t]) = \Phi_t \hat{\beta}[t] - \mathbf{r}_t, \quad (16)$$

$B(\hat{\beta}[t], \beta) = \|\hat{\beta}[t] - \beta\|_2^2$ is the Bregman divergence chosen in such a way that (15) has a closed-form solution, and η_t is the step size of the updates. In an online setting, Φ_t and $\mathbf{r}_t$ in (16) can be updated recursively as

$$\Phi_t = \gamma \Phi_{t-1} + \delta \mathbf{F}_t^\top (\mathbf{I} + \mu_1 \mathbf{L}_{1,\ell} + \mu_2 \mathbf{L}_{1,u}) \mathbf{F}_t, \quad (17)$$

$$\mathbf{r}_t = \gamma \mathbf{r}_{t-1} + \delta \mathbf{F}_t^\top \mathbf{f}_t. \quad (18)$$

Using the notation $\mathbf{g}[t] = [\mathbf{g}_1[t]^\top, \dots, \mathbf{g}_P[t]^\top]^\top := \nabla \ell_t(\hat{\beta}[t])$ and the chosen Bregman divergence, (15) can be written as

$$\hat{\beta}[t+1] = \arg \min_{\beta} \frac{\|\beta\|_2^2}{2\eta_t} + \beta^\top (\mathbf{g}[t] - \frac{\hat{\beta}[t]}{\eta_t}) + \omega(\beta). \quad (19)$$

Note that (19) is separable in p ; solving for each $\hat{\beta}_p[t+1]$ leads to

$$\hat{\beta}_p[t+1] = \arg \min_{\beta_p} \frac{\|\beta_p\|_2^2}{2\eta_t} + \beta_p^\top [\mathbf{g}_p[t] - \frac{\hat{\beta}_p[t]}{\eta_t}] + \lambda \|\beta_p\|_2 \quad (20)$$

A closed-form solution of (20) can be obtained using the multidimensional shrinkage thresholding operator [25]:

$$\hat{\beta}_p[t+1] = (\hat{\beta}_p[t] - \eta_t \mathbf{g}_p[t]) \left[1 - \frac{\eta_t \lambda}{\|\hat{\beta}_p[t] - \eta_t \mathbf{g}_p[t]\|_2} \right]_+, \quad (21)$$

where $[x]_+ = \max(x, 0)$. In (21), the first factor is the typical gradient descent update, and the second enforces group sparsity. While our proposed online topology estimator builds upon existing COMID algorithm, it introduces novel constraints based on the simplicial structure that have not been explored in existing literature.

The computational complexity of the algorithm is mainly contributed by (16) and (17), and it is of order $\mathcal{O}(K^2 P^2 + N_1^2 K P)$. We remark that under certain assumptions, similarly as in [4], a sublinear dynamic regret that guarantees the convergence of (21) can be derived, which is not reported here due to space limitations.

4. NUMERICAL RESULTS

We test the S-VAR to forecast edge flows on traffic data from Sioux Falls transportation network [26] and water flow data from Cherry Hills water network [27]. The performance accuracy is evaluated using the normalized mean squared error (NMSE), defined as

$$\text{NMSE}(T) = \frac{1}{N_1} \sum_{e=1}^{N_1} \frac{\sum_{t=1}^T (f_e(t) - \hat{f}_e(t))^2}{\sum_{t=1}^T f_e(t)^2}, \quad (22)$$

where $\hat{f}_e(t)$ denotes the predicted flow on edge e at time t . As

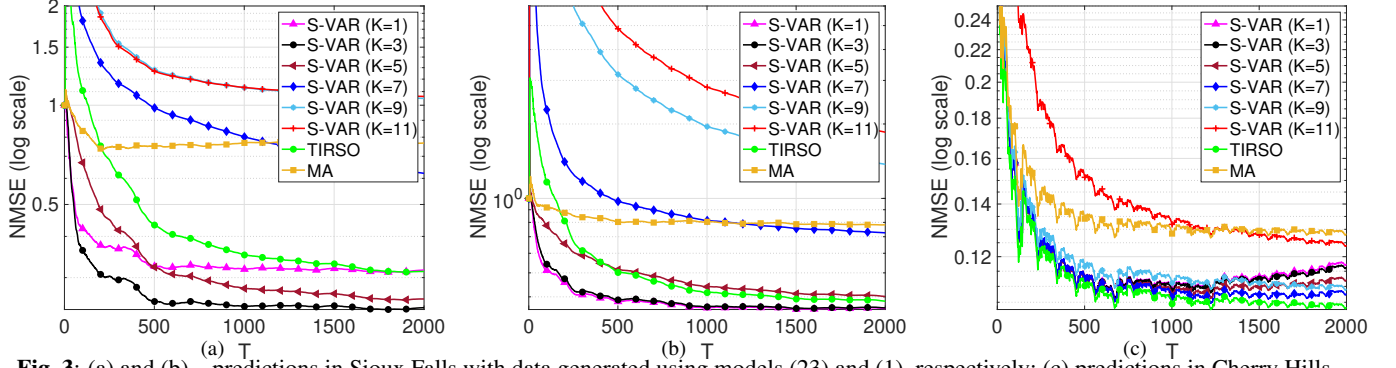


Fig. 3: (a) and (b)—predictions in Sioux Falls with data generated using models (23) and (1), respectively; (c) predictions in Cherry Hills.

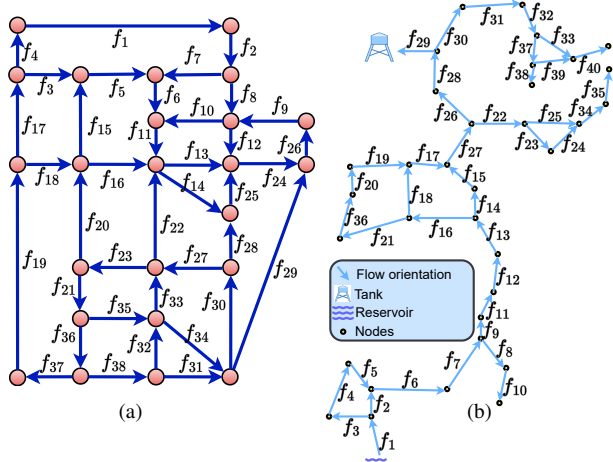


Fig. 4: (a) Sioux Falls transportation network, (b) Cherry Hills flows.

benchmarks, we consider the *Topology Identification via Recursive Sparse Online Optimization (TIRSO)* algorithm [4]—a state-of-the-art online algorithm to learn the vanilla VAR model—and the Moving Average (MA) algorithm. For the S-VAR, we show results for $K \in \{1, 3, 5, 7, 9, 11\}$, and in all the experiments, we assumed $K_\ell = K_u$ for the legibility of the presentation. A total of 3000 data samples are generated, and the first 1000 samples are used to tune the hyperparameters of all the algorithms using a grid search for the lowest NMSE, resulting in $(\mu_1, \mu_2, \lambda, \gamma) = (0.01, 0.001, 0.01, 0.98)$. Following the analysis in [4], we choose the step size $\eta_t = 1/\Lambda_{\max}(\Phi_t)$, where $\Lambda_{\max}(\Phi_t)$ denotes the maximum eigenvalue of Φ_t .

4.1. Sioux Falls Transportation Network

The Sioux Falls transportation network has 24 nodes (0-simplices), 38 edges (1-simplices), and 2 triangles (2-simplices), as shown in Fig. 4a [26]. The time-evolving edge signals are generated by assuming two different nonstationary VAR models.

First, we consider an inverse model as in (5), i.e.,

$$\mathbf{f}_t = \sum_{p=1}^3 \left(\sum_{k=0}^3 \beta_{p,k}^\ell \mathbf{L}_{1,\ell}^k + \sum_{k=q}^2 \beta_{p,k}^u \mathbf{L}_{1,u}^k \right) \mathbf{f}_{t-p} + \varepsilon_t, \quad (23)$$

where $\varepsilon_t \sim \mathcal{N}(0, 1)$. The filter coefficients $\beta_{p,k}^\ell$ and $\beta_{p,k}^u$ are randomly drawn from $\mathcal{N}(0, 1)$; and at every 100-th time step, all the coefficients are changed to make the setting more dynamic. Second, we consider a pure VAR model as in (1) of order $P=3$, where $\mathbf{W}_p$ is a random matrix with entries drawn from $\mathcal{N}(0, 1)$, and at every 100-th time step, all the coefficients are also changed; and $\varepsilon_t \sim \mathcal{N}(0, 1)$.

Table 1: Number of parameters: TIRSO vs best S-VAR

	TIRSO	S-VAR
Sioux Falls ($K = 3, P = 3, E = 38$)	$38 * 38 * 3 = 4332$	$3 * 3 = 9$
Cherry Hills ($K = 7, P = 2, E = 40$)	$40 * 40 * 2 = 3200$	$7 * 2 = 14$

The prediction results for both cases are plotted in Figs. 3a and 3b with $P = 3$, chosen using grid search. For both cases, S-VAR with $K \in \{1, 2, 3\}$ outperforms the benchmarks with a significant margin most of the time. The S-VAR lags behind the other algorithms for $K > 3$, since the model forces spatial interaction among the edges that are beyond the true underlying proximity. Table 1 compares the number of parameters between the best performing S-VAR ($K = 3$) and TIRSO, where it can be observed that the S-VAR has three orders of magnitude fewer. This may also be advantageous when the available data is limited.

4.2. Cherry Hills Water Network

Cherry Hills is a water network consisting of 36 nodes (0-simplices), 40 pipes (1-simplices), and 2 triangles (2-simplices) [27]. We assume a reference flow direction as in Fig. 4b, and generate time-evolving flow signals using the EPANET software with a demand-driven model such that the water flows meet the time-varying water demands at the nodes. The flow signals are the hourly sampled volume of water in m^3/h . The prediction NMSEs of all algorithms using a filter order $P = 2$ (chosen via grid search) are plotted in Fig. 3c. S-VAR achieves very close performance to TIRSO for $K = 7$, which showcases the ability of S-VAR to compete with the vanilla VAR with a remarkably less number of parameters (see Table 1). This is possible because of the simplicial inductive bias the S-VAR leverages, while TIRSO does not.

5. CONCLUSION

We proposed a simplicial VAR model, capturing the spatio-temporal dependencies among the signals defined over simplicial structures. By leveraging the simplicial convolution filters, the proposed model uses the simplicial structure as the inductive bias and exploits the structure-aware proximities of the network, resulting in significantly less number of parameters compared to the vanilla VAR model. Considering the real-world nonstationary networks, we developed an online learning strategy for the model based on a group-lasso optimization framework, solved using composite objective mirror descent. Numerical experiments with traffic and water networks demonstrated that the proposed method with a remarkably low number of parameters is quite competitive with the vanilla VAR model.

6. REFERENCES

- [1] H. Weiyu, G. Leah, W. Nicholas, G. Scott, B. Danielle, and R. Alejandro, "Graph frequency analysis of brain signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 7, 2016.
- [2] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognition*, vol. 121, 2022.
- [3] Rohan Money, Joshin Krishnan, and Baltasar Beferull-Lozano, "Online joint nonlinear topology identification and missing data imputation over dynamic graphs," *European Signal Processing Conference (EUSIPCO)*, 2022.
- [4] B. Zaman, L. M. L. Ramos, D. Romero, and B. Beferull-Lozano, "Online topology identification from vector autoregressive time series," *IEEE Transactions on Signal Processing*, vol. 69, pp. 210–225, 2021.
- [5] R. Money, J. Krishnan, and B. Beferull-Lozano, "Online nonlinear topology identification from graph-connected time series," in *2021 IEEE Data Science and Learning Workshop (DSLW)*, 2021, pp. 1–6.
- [6] R. Money, J. Krishnan, and B. Beferull-Lozano, "Random feature approximation for online nonlinear graph topology identification," *IEEE MLSP*, 2021.
- [7] R. Money, J. Krishnan, and B. Beferull-Lozano, "Sparse Online Learning with Kernels using Random Features for Estimating Nonlinear Dynamic Graphs," *10.36227/techrxiv.19210092.v3*, 8 2022.
- [8] Y. Shen, G. Giannakis, and B. Baingana, "Nonlinear structural vector autoregressive models with application to directed brain networks," *IEEE Transactions on Signal Processing*, vol. 67, pp. 5325–5339, 2019.
- [9] M. Veedu, D. Harish, and M. Salapaka, "Topology learning of linear dynamical systems with latent nodes using matrix decomposition," *IEEE Transactions on Automatic Control*, 2021.
- [10] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, "Forecasting time series with varma recursions on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4870–4885, 2019.
- [11] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, "Forecasting time series with VARMA recursions on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4870–4885, 2019.
- [12] M. Schaub, Y. Zhu, J. Seby, T. Roddenberry, and S. Segarra, "Signal processing on higher-order networks: Livin' on the edge... and beyond," *Signal Processing*, vol. 187, pp. 108149, 2021.
- [13] G. Leus, M. Yang, M. Coutino, and E. Isufi, "Topological volterra filters," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5385–5399.
- [14] G. Essl, "Topological IIR filters over simplicial topologies via sheaves," *IEEE Signal Processing Letters*, vol. 27, pp. 1215–1219, 2020.
- [15] S. Ebli, M. Defferrard, and G. Spreemann, "Simplicial neural networks," in *TDA & Beyond*, 2020.
- [16] 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.
- [17] E. Isufi and M. Yang, "Convolutional filtering in simplicial complexes," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 5578–5582.
- [18] L. Lim, "Hodge laplacians on graphs," *Siam Review*, vol. 62, no. 3, pp. 685–715, 2020.
- [19] S. Barbarossa and S. Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2992–3007, 2020.
- [20] M. Yang, E. Isufi, M. Schaub, and G. Leus, "Simplicial convolutional filters," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4633–4648, 2022.
- [21] A. Natali, E. Isufi, M. Coutino, and G. Leus, "Learning time-varying graphs from online data," *IEEE Open Journal of Signal Processing*, vol. 3, pp. 212–228, 2022.
- [22] D. Shuman, S. 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, May 2013.
- [23] M. Yang, E. Isufi, M. Schaub, and G. Leus, "Finite impulse response filters for simplicial complexes," in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 2005–2009.
- [24] J. Duchi, S. Shwartz, and A. Tewari, "Composite objective mirror descent," in *COLT'10*, 2010, pp. 14–26.
- [25] A. Puig, A. Wiesel, and A. Hero, "A multidimensional shrinkage-thresholding operator," *Proceedings of the 15th Workshop on Statistical Signal Processing*, vol. 18, pp. 113 – 116, 10 2009.
- [26] L. Leblanc, "An algorithm for the discrete network design problem," *Transportation Science*, vol. 9, no. 3, pp. 183–199, 1975.
- [27] L. Rossman, R. Clark, and W. Grayman, "Modeling chlorine residuals in drinking water distribution systems," *Journal of Environmental Engineering*, vol. 120, no. 4, pp. 803–820, 1994.