Lifting Scheme for Data in River Networks

Seoncheol Park
Pacific Climate Impacts Consortium
University of Victoria
Victoria, BC V8W 2Y2, Canada

Hee-Seok Oh
Department of Statistics
Seoul National University
Seoul 08826, Korea

Draft: version of March 16, 2020

Abstract

This paper presents a new multiscale method for analyzing water pollutant data located in river networks. The main idea of the proposed method is to adapt the conventional lifting scheme, one of the second-generation wavelets, reflecting the characteristics of streamflow data in the river network domain. Due to the complexity of the data domain structure, it is difficult to apply the lifting scheme to the streamflow data directly. To solve this problem, we propose a new lifting scheme algorithm for streamflow data that incorporates flow-adaptive neighborhood selection, flow proportional weight generation, and flow-length adaptive removal point selection. A non-decimated version of the proposed lifting scheme is also provided. The simulation study demonstrates that the proposed method successfully performs a multiscale analysis of streamflow data. Furthermore, we provide a real data analysis of water pollutant data observed on the Geum-River basin compared to the existing smoothing method.

Keywords: Lifting scheme; River network; Smoothing; Spatial adaptation; Spatial modeling; Streamflow data.

1 Introduction

Environmental monitoring is a collection of observations and studies for the evaluation of environmental data (Artiola *et al.*, 2004). Humans now know that the environment is crucial to our health and survival. Therefore, we cannot overemphasize environmental monitoring for humans. One of the main areas of environmental monitoring is water quality management. As human activities increase, more environmental costs are needed to rehabilitate water. Therefore, it is important to analyze the characteristics of water pollutants.

This paper focuses on the environmental pollutant called Total Organic Carbon (TOC, mg/L). Recently, the Korean Ministry of Environment announced that they changed the water pollution index for monitoring wastewater treatment performance of facilities from chemical oxygen demand (COD) to TOC. According to the ministry, it cannot measure all organic matters in the water. However, using TOC can compensate for these shortcomings. Therefore, analyzing TOC data is meaningful to society. The National Institute of Environmental Research (NIER) under the Ministry of Environment operates a water environment information system to monitor water quality. This system provides "Environment standard", which is a good guideline for the amount of TOC listed in Table 1.

Table 1: Environment standard for TOC provided by Water Environment Information System (WEIS).

Status	Very good	Good	Slightly better	Normal	Poor	Bad	Very bad
TOC (mg/L)	≤ 2	≤ 3	≤ 4	≤ 5	≤ 6	≤ 8	> 8

From Figure 1, we can find some characteristics of the water quality index: (i) The TOC index is located in the river network. It means that observations are correlated across the river network, not the usual $\mathbb{R}^2$ domain. Most statistical models are interested in analyzing a spatial region, a subset of $\mathbb{R}^2$, where Euclidean distance works well. On the other hand, for the streamflow data in Figure 1, Euclidean distance does not work well as a natural metric. (ii) As shown in Figure 1, the data have spatially inhomogeneous features in various dependent structures along with the river network. (iii) The data are irregularly observed in the river network.

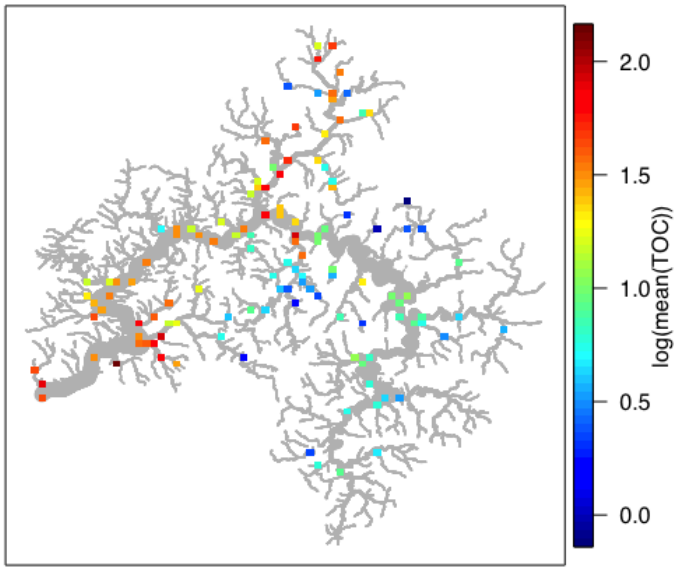

Figure 1: TOC data on the Geum-River network. Gray lines mean streamflow segments with different weights represented by their widths, and colored points mean logarithm values of TOC means from December 2011 to November 2017 in 127 observation sites.

Therefore, any such method of representing the above data should have the following features: (i) It is capable of effectively represent streamflow data in the river network. (ii) It provides a spatially adaptive framework to estimate the inhomogeneous underlying function by reflecting the inherent multiscale characteristics of data. (iii) It applies to scattered data in the river network. In this paper, we would like to propose a multiscale method that satisfies all of the features mentioned above.

In this paper, we assume that observations are a set of scattered data (x_i, y_i), $i = 1, \ldots, n$, as shown in Figure 1, from model

$$y_i = g(x_i) + \varepsilon_i, \tag{1}$$

where x_i denotes the locations of observations in the data domain, ε_i are the measurement errors, and g denotes an unknown underlying function of interest. Our goal is to estimate the underlying field $g(x)$ for every location x on the river network. In other words, for the case in Figure 1, we would like to represent the underlying field of the water quality index in the river network domain.

In the literature, there exist some studies of streamflow data analysis. VerHoef *et al.* (2006) proposed the use of stream distance defined by the shortest distance between two locations on river networks, as a reasonable distance measure for data analysis on the river network. They showed that it could construct a large class of valid spatial autocovariance models using the stream distance. They also suggested a kernel convolution-based method generates a class of covariance models for streamflow data. O'Donnell *et al.* (2014) used nonparametric flexible regression approaches, such as kernel methods and penalized splines, to build spatio-temporal models in river networks. They suggested a piecewise simple regression analysis by dividing the network into a large number of small pieces called "stream segments". They provided regression-based stream data estimates assuming that the function values g's are constant within the same stream segments.

Meanwhile, due to the complexity of the river network data, it is not easy to fully understand the underlying structure of the data. The multiscale analysis is a possible way to solve such a problem by considering the multiresolution of data. As conventional multiscale methods, wavelets are the most popular choice. However, wavelets do not properly work when the data are not observed on regular grids, or the number of observations is not dyadic, i.e., $n = 2^J$, for some $J \in \mathbb{Z}$. To overcome these problems, Sweldens (1996) and Sweldens (1998) proposed a kind of the second-generation wavelet called "lifting scheme". The lifting scheme has been extensively studied in signal processing and image analysis (Jansen and Oonincx , 2005).

However, there is a limit that all of the previous works cannot provide a multiscale structure for river network data. As far as we know, there is no direct literature describing multiscale methods for river network data. In this paper, a new lifting method for river network data is proposed by combining the conventional lifting method and novel modifications of neighborhood selection, filter prediction, and removal point selection, taking into account the characteristics of the data. The proposed method has two advantages. First, by following the argument of the lifting scheme, it gives a multiscale structure of river network data. Second, the proposed method is advantageous compared to the conventional smoothing methods for river networks from the signal denoising point of view.

The rest of the paper is organized as follows. Section 2 reviews the existing lifting schemes

and smoothing method in the river network. Section 3 describes the river network data used in this study. Section 4 presents a new method termed "streamflow lifting scheme". Simulation studies and real data analysis are conducted in Sections 5 and 6 to evaluate the proposed method. Finally, concluding remarks are provided in Section 7.

2 Backgrounds

2.1 Lifting scheme

In this section, we briefly summarize the concept of lifting scheme for self-contained material. Suppose that we observe a set of n irregular locations $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, where the length of the data ($= n$) may not be dyadic. Assume that we have function values $y_1, \ldots, y_n$ at every location. We want to construct a multiresolution transform at the $j-1$th level, given the jth level data $\boldsymbol{y}_j$. The lifting scheme consists of the following four steps:

1. **Split**: At the level $j-1$, divide $\boldsymbol{y}_j$ into two subsets, $\mathcal{P}_{j-1}$ and $\mathcal{U}_{j-1}$.

2. **Predict**: Predict every sample $y_{j,i} \in \mathcal{P}_{j-1}$ from $y_{j,k} \in \mathcal{U}_{j-1}$ with a prediction filter $\mathbf{p}_{j-1,i}$, and store the prediction error $d_{j-1,i} = y_{j,i} - \hat{y}_{j,i} = y_{j,i} - \sum_{k \in \mathcal{N}_{j-1,i} \cap \mathcal{U}_{j-1}} p_{j-1,i,k} y_{j,k}$, where $\mathcal{N}_{j-1,i}$ is the set of neighbors of node i, $\hat{y}_{j,i}$ represents the predicted value constructed from $\mathcal{U}_{j-1}$ neighbors of node i. Note that i and k denote location in $\mathcal{P}_{j-1}$ and $\mathcal{U}_{j-1}$, respectively.

3. **Update**: Update the $j-1$th level data $y_{j-1,k}$ in $\mathcal{U}_{j-1}$ with an appropriate update filter $\mathbf{u}_{j-1,k}$, that is, $y_{j-1,k} = y_{j,k} + \sum_{i \in \mathcal{N}_{j-1,k} \cap \mathcal{P}_{j-1}} u_{j-1,k,i} d_{j-1,i}$.

4. **Repeat**: Repeat the above steps until the desired resolution level is achieved.

By performing these steps, we construct coarse signals of data from updated subsamples. Meanwhile, the reverse version of the lifting scheme can be easily obtained by undoing the forward scheme operations at the level $j-1$: (i) Undo update: $y_{j,k} = y_{j-1,k} - \sum_{i \in \mathcal{N}_{j-1,k} \cap \mathcal{P}_{j-1}} u_{j-1,k,i} d_{j-1,i}$. (ii) Undo predict: $d_{j,i} = d_{j-1,i} + \sum_{k \in \mathcal{N}_{j-1,i} \cap \mathcal{U}_{j-1}} p_{j-1,i,k} y_{j,k}$. (iii) Undo split. (iv) Repeat the above steps at the next level.

There are several crucial components in the construction of the lifting scheme.

- **The number of removing points at ones** ($|\mathcal{P}|$): The user should select how many points remain at the next (coarser) level.

- **Prediction filter**: It is essential to choose a prediction filter in the procedure. Haar (local constant), local linear, local polynomial, or inverse distance weight are frequently used in the literature.

- **Removal order of points**: When removing the points, it is crucial to determine the order in which points are removed. It relates to the question of whether or not what is essential to represent the underlying field.

- **Neighborhood selection**: It is important to select multiple neighbors to construct a prediction filter. Too many neighbors make it difficult to understand the local behavior of the data, while too few neighbors yield a bias to predict each node.

2.1.1 Lifting one coefficient at a time (LOCAAT)

We now review the lifting one coefficient at a time (LOCAAT) algorithm of Jansen *et al.* (2009). The LOCAAT algorithm constructs a removal order of data points and sequentially decomposes the data with the order. Suppose we have the values $y_1, \ldots, y_n$, sampled at n irregularly spaced points $x_1, \ldots, x_n$ on the real line. Lifting scheme approximates the function g in (1) as

$$\tilde{g}(y) = \sum_{k=1}^{n} c_{n,k}\phi_{n,k}(x),$$

where $c_{n,i} := g(x_i)$, $\phi_{n,k}(x_i) = \delta_{i,k}$ for $k, i \in \{1, \ldots, n\}$, and $\delta_{i,k}$ denotes the Kronecker delta.

The LOCAAT algorithm first defines the index set of the scaling coefficients as $\mathcal{U}_n = \{1, \ldots, n\}$ and the index set of wavelet coefficients as $\mathcal{P}_n = \emptyset$. At the next step $n-1$, a point to be lifted is selected and denoted as j_n, which is the point to be removed from the current set of scaling coefficients and to be converted into a detailed coefficient. The new set of indices corresponding to the scaling coefficients is $\mathcal{U}_{n-1} = \mathcal{U}_n \backslash \{j_n\}$, while $\mathcal{P}_{n-1} = \{j_n\}$ is the index set of the wavelet coefficient constructed at this stage.

To select the point to be lifted, Jansen *et al.* (2009) used the minimum of the integral of scaling function $\phi_{n,k}$, I_{nk}, as an appropriate measure. In this study, length or volume is considered as an appropriate measure. Figure 2 shows a toy example that selects a point

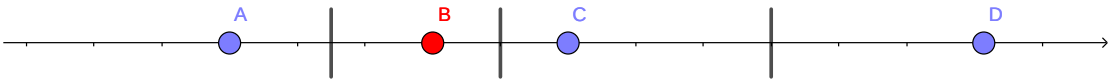

Figure 2: Illustration of the removal point selection of lifting scheme in the real line.

to be lifted by the LOCAAT algorithm. A, B, C, and D denote locations. The area of each point can be defined by dividing the real line into four blocks using the midpoint marked by the vertical line. The LOCAAT algorithm selects among the candidates a point with the smallest area corresponding to the length of each block in the one-dimensional data area. In this example, a point of B is selected to be removed. For the configuration of update filters, Jansen *et al.* (2009) proposed a minimum norm solution-based update weights at level r due to the numerical stability,

$$b_j^r = I_{ri_r} I_{r-1,j} / \sum_{k \in \mathcal{N}_r} I_{r-1,k}^2, \tag{2}$$

where i_r is an index of the candidate points for removal.

2.1.2 Other lifting schemes

Nunes *et al.* (2006) proposed a new lifting scheme called "adaptive lifting". The key ingredients of the adaptive lifting are the data-adaptive selection of the removal order and the neighborhood size in the lifting prediction step. Especially, they considered linear, quadratic, and cubic regression-based prediction filter in each step in LOCAAT and selected a filter which generates the smallest absolute value of detail coefficient. In the same way, we can decide the optimal neighborhood size of each predict step. Through these modifications, Nunes *et al.* (2006) flexibly constructed prediction filters in the one-dimensional signal denoising setting.

In the lifting scheme, it is not easy to find the optimal removal sequence because there is no optimal removal order in terms of minimizing the mean square error. To enhance the performance of the lifting scheme in nonparametric regression settings, Knight *et al.* (2009) proposed a "nondecimated" concept in lifting transform. It borrows the idea from a nondecimated wavelet transform that uses over-determined basis functions to improves the performance of the wavelet transform. Knight *et al.* (2009) generated several removal order sequences called paths. They generated Q different removal orders by permutation. Following notations of Knight *et al.* (2009), let $\hat{g}^{(q)}(x)$ be the estimate of the unknown function g at

locations x, using the qth path. They showed that an averaged estimator

$$\hat{\bar{g}}(x_i) = \frac{1}{Q}\sum_{q=1}^{Q} \hat{g}^{(q)}(x_i), \qquad \forall i = 1, \ldots, n$$

could reduce the error between the true signal and its estimator. For a better smoothing, Knight *et al.* (2009) further selected a few paths that provide lower approximated average square errors $\widehat{\text{ASE}}$,

$$\widehat{\text{ASE}}(\hat{g}^{(q)}, g) = \frac{1}{n}\sum_{i=1}^{n} \{\hat{g}^{(q)}(x_i) - \hat{\bar{g}}(x_i)\}^2.$$

2.2 Shrinkage in the lifting scheme

Lifting schemes have also been applied to nonparametric regression problems by incorporating a shrinkage approach. The main idea of shrinkage is based on the assumption that the true signal information is contained only in large values of the $\mathbf{d}$ elements. Thus, by setting the d coefficient less than a specific threshold to zero, the reconstruction results may be more similar to the true signal.

In the proposed streamflow lifting scheme to be discussed in Section 4, we use the same shrinkage strategies used in Nunes *et al.* (2006) and Knight *et al.* (2009). There are several types of shrinkage approaches. In this paper, we focus on the median and hard thresholds, which are implemented as `median` and `hard` in `adlift` and `nlt` package in R. To use the lifting scheme, one must decide the number of scaling coefficients to be kept in the final representation of the initial signal. The user also specifies `nkeep` in `adlift` and `nlt` package in R. In this paper, we use the fully decomposed result (`nkeep=2`) in Knight *et al.* (2009), which produces $(n-2)$ detail coefficients in the length-n dataset.

2.3 Smoothing method on river networks

In this subsection, we briefly summarize the work of O'Donnell *et al.* (2014). One of the key ideas of O'Donnell *et al.* (2014) is to simplify the information in a given network using the concept of stream segments. They also suggested a penalize spline-based method with spatial, seasonal, temporal, and interaction bases. This paper focuses on the analysis of the spatial behavior of pollutants, taking into account the structure of river networks. For this

purpose, we consider a straightforward spatial additive model as

$$y_i = \mu + m_x(x_i) + \varepsilon_i = g(x_i) + \varepsilon_i, \tag{3}$$

where m_x describes spatial trends. The main idea of the spline method is to use a set of basis functions to estimate g in (1). So, with p basis functions, the estimator is $\hat{g}(x) = \sum_{j=1}^{p} \beta_j \phi_j(x)$. O'Donnell *et al.* (2014) used a P-spline basis, which is a penalized version of a B-spline basis. More specifically, a B-spline model is formulated as $\mathbf{y} = B\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $B = (1, B_s)$, where B_s is a design matrix of spatial components, and $\boldsymbol{\beta}$ is an $n \times p$ response vector. The model is fitted by minimizing the following penalized sum of squares

$$(\mathbf{y} - B\boldsymbol{\beta})^T(\mathbf{y} - B\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^T D^T D \boldsymbol{\beta}, \tag{4}$$

where D denotes the penalty matrix. The solution of (4) is $\hat{\boldsymbol{\beta}} = (B^T B + \lambda D^T D)^{-1} B^T \mathbf{y}$, where λ is a smoothing parameter. For the optimal value of λ, O'Donnell *et al.* (2014) selected λ to minimize $\log(\hat{\sigma}^2) + 1 + \frac{2+2\text{df}}{n-\text{dof}-2}$, where df denotes the degree of freedom. For the detailed information of smoothing methods in the stream network. refer to O'Donnell *et al.* (2014).

3 Geum-River TOC data

The data used in this paper are observed in the Geum-River basin located in the heart of South Korea. See Figure 3(a). According to the Water Environment Information System operated by the Ministry of Environment, the Geum-River basin is divided by 14 sub-regions, called catchments, which are plotted with solid lines in Figure 3(a) and (b). All 14 catchments are also divided into several sub-catchments, which are plotted with dotted lines in Figure 3(a) and (b). Among them, the Miho-Cheon catchment marked by orange in Figure 3(b) is one of the sub-regions. It contains many observational stations compared to other catchments, and there are several cities and factories around it. We believe it is meaningful to take a closer look at the area. This river network is also applied to build a network model for simulation studies in Section 5.

The orange lines in the Miho-Cheon catchment of Figure 3(b) represent stream segments defined by lines between junctions of the river network (VerHoef *et al.*, 2006, 2010). We note that there are 113 strea‗m segments and 28 observation stations in the Miho-Cheon

catchment. The Geum-River network has a total of 942 stream segments and 127 observation points.

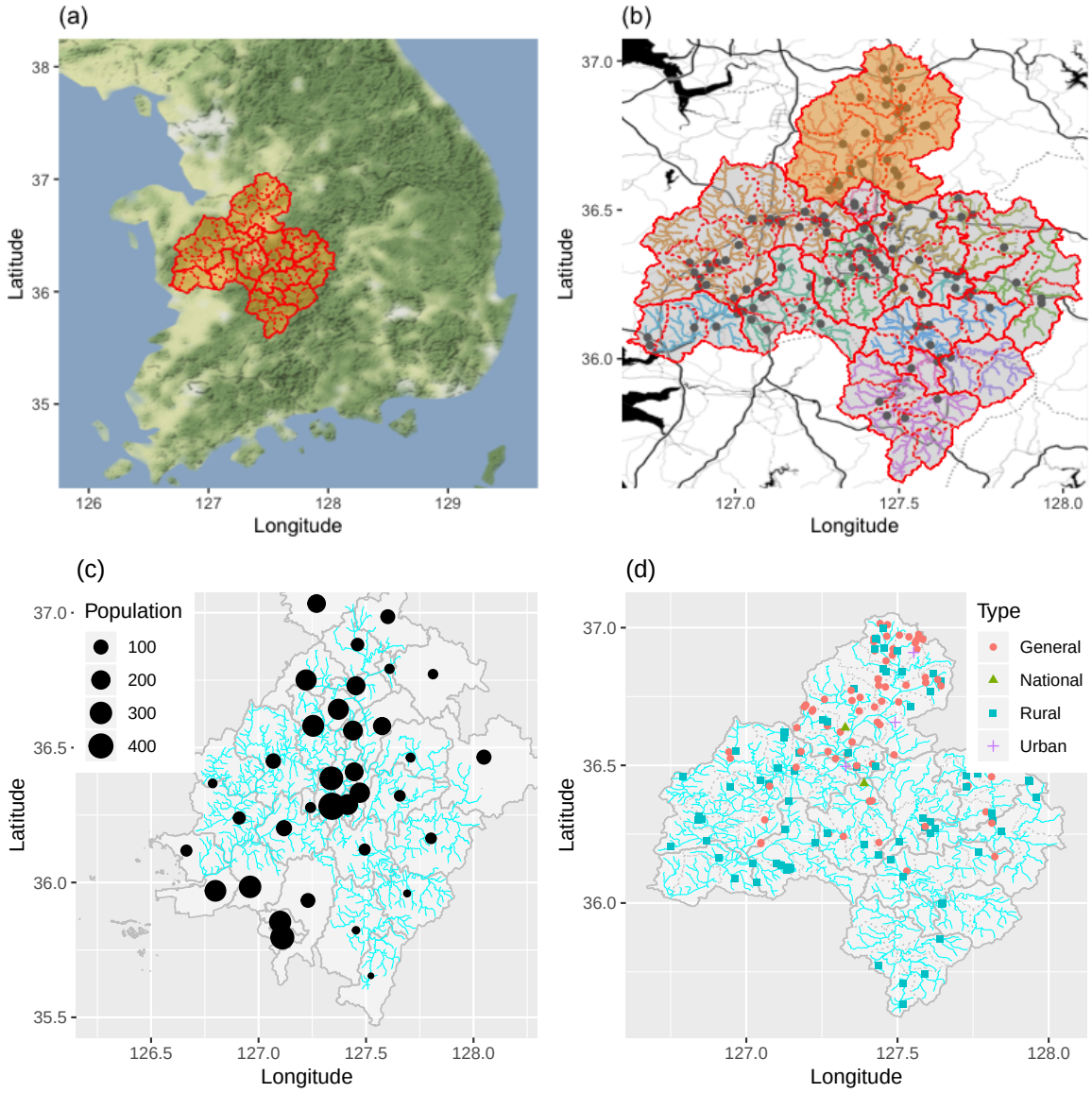

Figure 3: (a) The Geum-River catchments in South Korea marked by red lines. (b) The Enlarged figure of (a). Black dots are 127 observation points. (c) Populations (2017/12/31, thousands) in the Geum-River basin. (d) Locations of industrial areas in the Geum-River basin.

Figure 3(c) shows the cities, counties, and districts populations located in the Geum-River

basin. Note that these administrative areas do not fully match the Geum-River catchments. From Figure 3(c), we observe that most of the populations are concentrated in the Northern and Central parts of the Geum-River basin. Figure 3(d) shows the locations of industrial areas in Geum-River basins. Note that general, national, and urban industrial sites are clustered in the Miho-Cheon and its nearby areas. Therefore, it is possible to assume that many water pollutants will occur in the Miho-Cheon and its adjacent river basin.

4 Streamflow lifting scheme

This section presents our procedure to construct a new lifting scheme for streamflow data by modifying the LOCAAT algorithm of Jansen *et al.* (2009) to adapt some characteristics of streamflow data. Our main idea is to develop a multiscale method for streamflow data analysis by incorporating the idea of Nunes *et al.* (2006) into O'Donnell *et al.* (2014). The necessary modifications for developing the streamflow lifting scheme are as follows: (i) performing a network-adaptive neighborhood selection, (ii) constructing a prediction filter with flow-adaptive weighted averages, and (iii) determining a removal order by defining a proper contribution measure of each observation point to the river network.

We consider a toy example network shown in Figure 4. Suppose that there are five observation points (A, B, C, D, E) in the different stream segments of a river network. Assume that each segment has a flow volume of f. Let $f_A, f_B, \ldots, f_E$ denote the flow volume of station $A, B, \ldots, E$, respectively. We further denote $y_A, y_B, \ldots, y_E$ as water quality observations at station $A, B, \ldots, E$, respectively.

4.1 Neighborhood selection

The concept of "flow-connected" introduced in VerHoef *et al.* (2006) is useful to build a neighborhood set of a point in a river network. VerHoef *et al.* (2006) defined that two locations are connected when the intersection of upstreams of two stations is a non-empty set. In our example, segments A, C and B, C are "flow-connected" because the water in A and B can go to location C. On the other hand, C and D are not flow-connected since the water in C cannot go to station D or vice versa.

We use the concept of "flow-connected" to determine whether the two segments are

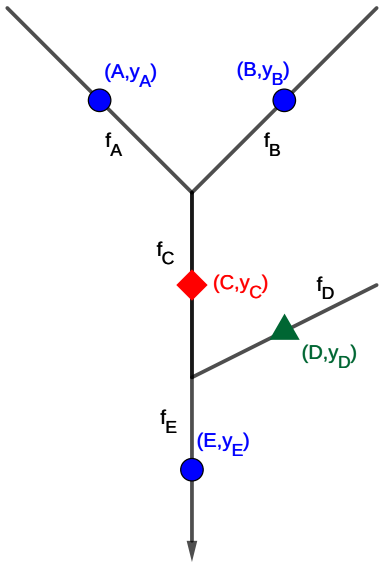

Figure 4: A simple example of streamflow data. Five solid black lines represent stream segments, indexed by A, B, C, D, and E. Each line segment has its flow volume, called $f_A, f_B, \ldots, f_E$. $y_A, \ldots, y_E$ denote water quality values of each segment.

neighbors or not. In this study, when two points are flow-connected, we consider each other neighbors. In Figure 4, suppose that we are interested in removing point C at a specific resolution level. By following the concept of flow-connected, A, B, and E (blue circles) are defined as its neighbors, and D (green triangle) is excluded from the neighborhood of C.

One of the distinct characteristics of the proposed neighborhood selection is that it considers both upstream and downstream neighborhoods. By doing so, it can reduce the number of boundary points. At first glance, including downstream points into the neighborhood seems awkward. However, by combining an appropriate prediction filter construction explained in Section 4.2, it can generate reasonable prediction filters.

4.2 Construction of prediction filter

In this section, we consider the problem of the prediction filter construction. The simplest prediction filter is constructed using an equally weighted value vector. However, every river network has its mainstream and sub-streams. It is plausible that observations on the main-

stream usually have a stronger effect on nearby observations. Therefore, the effect of each stream on a given segment should be different. To take into account the influence of each stream segment, we use flow volumes. For the construction of the prediction filter, we consider the size of flow volumes compared to others, called "relative flow volumes" (O'Donnell *et al.*, 2014).

Suppose that we have neighbors of a specific point in a river network. An easy way to weigh is to give the same weight to all neighbors, which may not be desirable. For example, if f_A is much larger than f_B, y_A has a more significant effect on y_C than y_B. Also, if f_D is larger than f_C, y_E is much different from y_C. Therefore, we intend to construct flow-adaptive weights that reflect the above considerations. We now consider predicting the response value of point C with the neighbors in the toy example in Figure 4. Since $f_C = f_A + f_B$, flow-adaptive weights for point C can be defined as ratios of flows,

$$w_A = \frac{f_A}{f_C}, \quad w_B = \frac{f_B}{f_C}, \quad \text{and} \quad w_E = \frac{f_C}{f_E}. \tag{5}$$

Then we obtain a predicted value of y_C as $\hat{y}_C = \tilde{w}_A y_A + \tilde{w}_B y_B + \tilde{w}_E y_E$, where

$$\begin{aligned} \tilde{w}_A &= \frac{f_A/f_C}{f_A/f_C + f_B/f_C + f_C/f_E}, \\ \tilde{w}_B &= \frac{f_B/f_C}{f_A/f_C + f_B/f_C + f_C/f_E}, \text{ and} \\ \tilde{w}_E &= \frac{f_C/f_E}{f_A/f_C + f_B/f_C + f_C/f_E}, \end{aligned} \tag{6}$$

which are normalized flow-adaptive weights to make the sum of weights to be 1, i.e., $\tilde{w}_A + \tilde{w}_B + \tilde{w}_E = 1$. Therefore, the predicted value of the segment C, $\hat{y}_C$ is

$$\hat{y}_C = \tilde{w}_A y_A + \tilde{w}_B y_B + \tilde{w}_E y_E.$$

Hence, we provide a lifting scheme for streamflow data by combining flow-adaptive weights of (5) and (6) with the conventional lifting scheme. In practice, it is rare to know all f values on the entire streamlines. Therefore, it is necessary to estimate flow values. For example, VerHoef *et al.* (2006) used equal weights for each split. In this study, it is assumed that the flow volume f in most upstream segments is proportional to their stream order and segment length. Note that the stream order is a positive whole number that is often used in hydrology to define stream-based distance in river networks. There are several stream

orders. Among them, the Shreve stream order is one of the most straightforward stream orders (Cressie *et al.*, 2006; VerHoef *et al.*, 2010). Cressie *et al.* (2006) defined the stream order as the number of sources in the upstream portion of the river network. The Shreve stream order starts from setting all most upstream segments to 1. Magnitudes increase at all junctions in the river network. For example, if a stream has a magnitude one and combines with a new stream having magnitude 2, it becomes magnitude 3. By doing so, it is able to configure all magnitudes of the given network.

To approximate f values, we use the Shreve stream order and assume that the flow of the most upstream segments is proportional to their lengths to prevent multiple tie values of flow volumes. After defining flow volumes of most upstream segments, one can define flow volumes of the next upstream segments as a sum of their upstream segments. By repeating this approach, we obtain all f values in the river network. It is also assumed that the weights associated with the flow volumes are known to generate $\log(\sqrt{f})$ values. Following O'Donnell *et al.* (2014), we normalize the $\log(\sqrt{f})$ values, which are between 0.2 to 1.5.

4.3 Removal point selection

The removal order should be determined for the streamflow lifting scheme. If the data lie in the real line, it is easy to apply the conventional approach, such as Nunes *et al.* (2006). They used the length of points on the real line for integral calculations. Moreover, it can be extended to the two-dimensional data proposed by Jansen *et al.* (2009). To determine the removal point, Jansen *et al.* (2009) found the highest density observation in the Euclidean domain by considering the integral of the scaling function. In addition, they proposed measuring the Voronoi-polygon based area as a candidate for proper integrals and chose to have the smallest integration point as a removal point in the LOCAAT algorithm.

However, these methods cannot be applied directly to streamflow data because the network is not easily projected into one- or two-dimensional data. In the streamflow lifting scheme, a simple approach is proposed to measure the contribution of each segment in the data to distinguish the points located in the most densest areas of the river network. We define an integral as the contribution of each observation point to the network. More specifically, to define the contribution of each point in streamflow data, we use flow-adaptive

weights defined in (6). A simple example is illustrated in Figure 4. Suppose that at the jth level, we want to remove point C with neighborhood points A, B, and E. Let I_A^j denote the integral of point A at the jth level, which is defined by the volume of the segment where A is located, say V_A,

$$\begin{aligned} I_A^j &= V_A = f_A \times \ell_A, \\ I_B^j &= V_B = f_B \times \ell_B, \text{ and} \\ I_E^j &= V_E = f_E \times \ell_E. \end{aligned}$$

At the next level $j-1$ after point C is removed, we need to update the integral of neighborhood points. For this purpose, we use a weighted volume of the point C according to the weights of neighbors in (6). Thus, I_A^j, I_B^j, and I_E^j are updated to

$$\begin{aligned} I_A^{j-1} &= I_A^j + \tilde{w}_A \times V_C, \\ I_B^{j-1} &= I_B^j + \tilde{w}_B \times V_C, \text{ and} \\ I_E^{j-1} &= I_E^j + \tilde{w}_E \times V_C. \end{aligned}$$

Note that since $\tilde{w}_A \times V_C + \tilde{w}_B \times V_C + \tilde{w}_E \times V_C = I_C^j$, the sum of integrals does not change. We select a point that has the minimum value of I^{j-1} for the removal point at the $j-1$th level. For the update filter, we use the minimum norm solution-based filter in (2).

4.4 Nondecimated lifting scheme for streamflow data

In this section, the proposed lifting scheme is generalized to a nondecimated version of the streamflow lifting scheme that can reduce the mean squared error of the lifting scheme in nonparametric regression settings, as mentioned in Section 2.1.2. For this purpose, we assume that the current stream distance-based removal order is one of the well-behaved trajectories in terms of the root mean squared error. Then we generate multiple trajectories from permutations. To generate these well-behaved trajectories, we first make clusters of observations and do permutation to those within the same cluster. For implementation, two tuning parameters should be used: (i) the number of trajectories (`Q=10`) and (ii) the number of permutation within a single trajectory (`v=5`).

5 Simulation study

This section conducts numerical experiments for the evaluation of our approach for streamflow data analysis. Assume that the data are observed from the regression model of 1. We mainly focus on the situation in which the underlying mean-field of the data is piecewise constant. Thus, there are several discontinuous function values in a river network, which may not be able to make proper estimates using conventional smoothing-based methods. For comparison, we consider the flexible smoothing approach of O'Donnell *et al.* (2014) and three variants of the proposed method: streamflow lifting scheme with median thresholding (`S-Lifting (M)`), streamflow lifting scheme with hard thresholding (`S-Lifting (H)`), and nondecimated streamflow lifting scheme with median thresholding (`S-Lifting (N)`).

For simulation setup, two types of river networks are considered: one is the Miho-Cheon streamflow segments in Figure 5(a) and (b), and the other is the simulated river network in Figure 5(c) and (d), which was used in Gallacher *et al.* (2017). The two networks consist of 113 stream segments and 80 stream segments, respectively. For each river network, the entire stream segments are divided into two groups: most upstream segments and non-most upstream segments, as shown in Figure 5 (a). Assume that there are no intrinsic sources to change the simulated signal values. The signal values in the non-most upstream segments are then generated from a weighted average of nearby upstream signal values. It implies that the simulation is sufficient to generate only the signal values in the most upstream segments.

In addition, we divide the most upstream segments into several clusters, as shown in red circles Figure 5 (a) and (c), to generate inhomogeneous stream network data. We assume that the signal values for all most upstream segments within the cluster are the same. For each simulated data set, $g(x_i)$ values of the most upstream segments are generated as follows: (i) All $g(x_i)$ values in the most upstream segments are set to be 9. (ii) A cluster is randomly selected from the clusters in Figure 5, and (iii) $g(x_i)$ values in the selected cluster are replaced with a value that is randomly chosen from $\{12, 15, 18\}$. This procedure is repeated until at least 30 most upstream segments have values greater than 9. A realization of the simulated data generation is shown in Figure 6 (a) and Figure 7 (a).

Three spatial sampling designs are also considered for simulation data in river networks. (i) For a sparse design, among a total of 113 Miho-Cheon stream segments, 40 stations

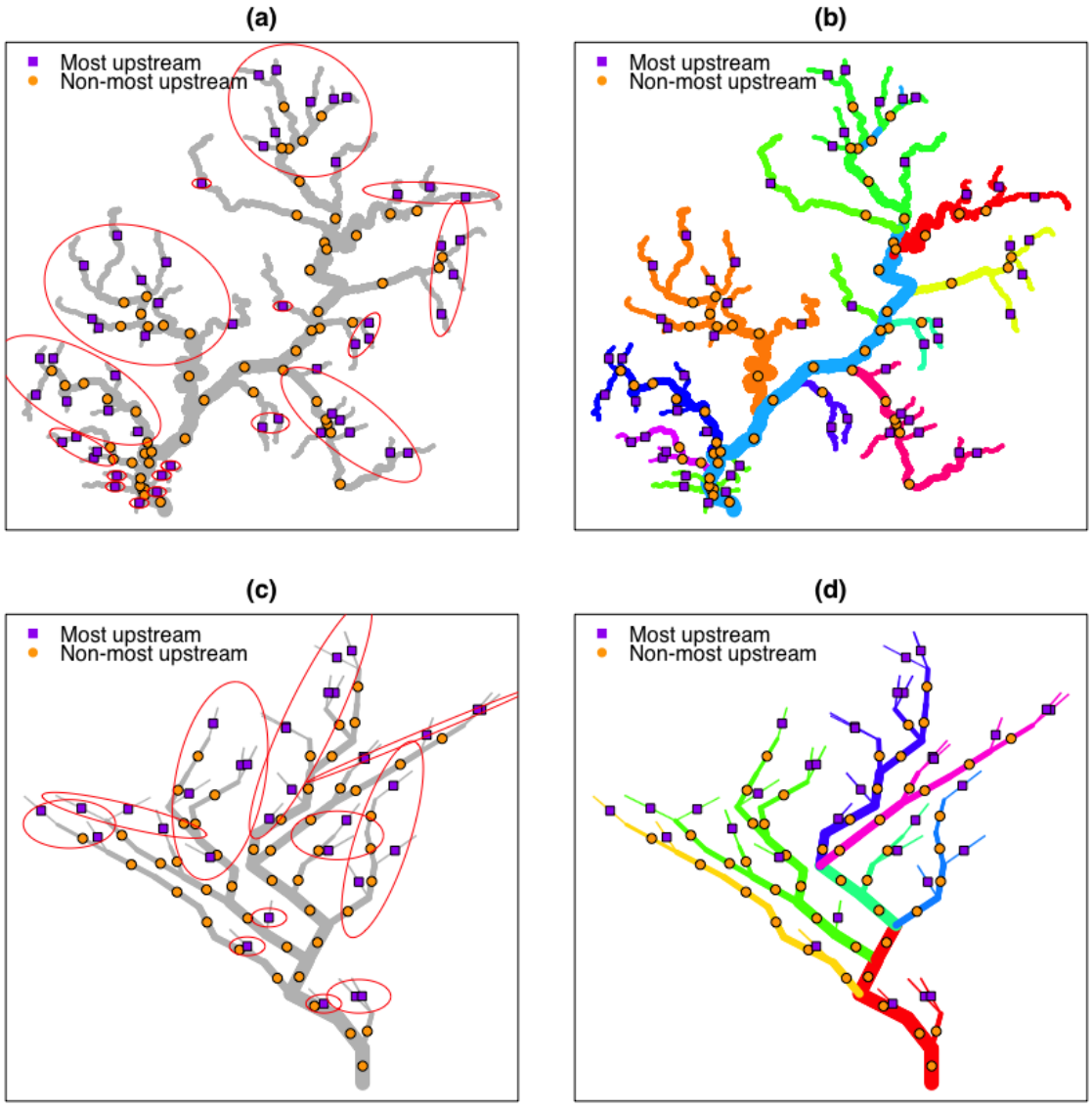

Figure 5: (a) Clusters (red circles) in the Miho-Cheon stream network. Note that the most upstream segments (purple squares) and the non-most upstream segments (orange circles). (b) Colors represent sub-streams for the sampling procedure. The sampling probability is proportional to the number of streams of each sub-stream. (c) and (d) show the same information of (a) and (b) for simulated river network used in Gallacher *et al.* (2017).

located on 40 different segments are considered. A realization is shown in Figure 5(b). (ii) 80 stations are used, which is nearly two-thirds of the number of the Miho-Cheon streams. (iii) 113 stations are considered as a dense case. Along the same line, we analyze the simulated network of Gallacher *et al.* (2017) in two designs: (i) observations are generated at 40 stations, and (ii) one observation is simulated in each segment. To selects stations in a river network, we use a spatial stratified sampling approach to make the resulting stations are evenly distributed in the network. Groups used in the sampling procedure are shown in Figure (b) and (d), respectively.

The noise terms are generated from $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.5$, $\sigma = 1$, and $\sigma = 1.5$. As for the evaluation measure, we consider the root mean square error (RMSE) as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_{tot}} (g(x_i) - \hat{g}(x_i))^2}{N_{tot}}},$$

where $\hat{g}(x_i)$ is an estimate of segment i, and N_{tot} denotes the total number of stream segments in the river network. For each combination of three spatial designs and three σ's, we compute RMSE values according to our methods and the approach of O'Donnell *et al.* (2014) over 100 simulated data sets.

Tables 2 and 3 list the averages of RMSE values under the two river networks. From results in the tables, we have some observations: (i) The proposed method outperforms the approach of O'Donnell *et al.* (2014) for most of the combinations. (ii) The proposed methods work well under the given simulation settings, especially when σ is small. (iii) The method by O'Donnell *et al.* (2014) provides stable results across the number of sampling observations and σ's, while the performance of the proposed methods is affected by both scenarios. (iv) The nondecimated version of the proposed lifting scheme is well performed in most cases. For visual inspection, we look at one realization example of the fitting results of the spatial design with 60 stations and $\sigma = 1$ shown in Figures 6 and 7. It seems that the proposed methods are well performed, reflecting the inhomogeneous features of the underlying fields in the two river networks.

We finally note that R codes used to implement the methods and to carry out some experiments are available at `https://github.com/SeoncheolPark/paper_StreamflowLifting/tree/master/code` in order that one can reproduce the same results.

Table 2: Averages of RMSE values (standard error) over 100 simulations.

RMSE (Std. error)	Obs=40			Obs=80			Obs=113		
	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
O'Donnell	2.0988 (0.1957)	2.2101 (0.2052)	**2.2882 (0.2317)**	1.5270 (0.1561)	1.6336 (0.1513)	1.7542 (0.1654)	1.3252 (0.1364)	1.4458 (0.1320)	1.5798 (0.1609)
Proposed (Median)	1.7089 (0.4181)	2.1410 (0.5202)	2.3588 (0.5134)	1.0662 (0.1877)	1.3488 (0.2343)	1.6377 (0.2686)	0.8362 (0.1187)	1.1656 (0.1929)	1.4141 (0.2119)
Proposed (Hard)	**1.6287 (0.3882)**	2.1281 (0.4913)	2.3647 (0.4704)	**1.0382 (0.1528)**	1.3440 (0.1903)	1.6400 (0.2524)	0.8386 (0.1074)	1.2142 (0.1569)	1.5172 (0.1985)
Proposed (Median, nlt)	1.7210 (0.3907)	**2.1143 (0.4517)**	2.3292 (0.4460)	1.0428 (0.1849)	**1.3212 (0.2241)**	**1.6076 (0.2588)**	**0.8000 (0.1158)**	**1.1225 (0.1760)**	**1.3705 (0.2081)**

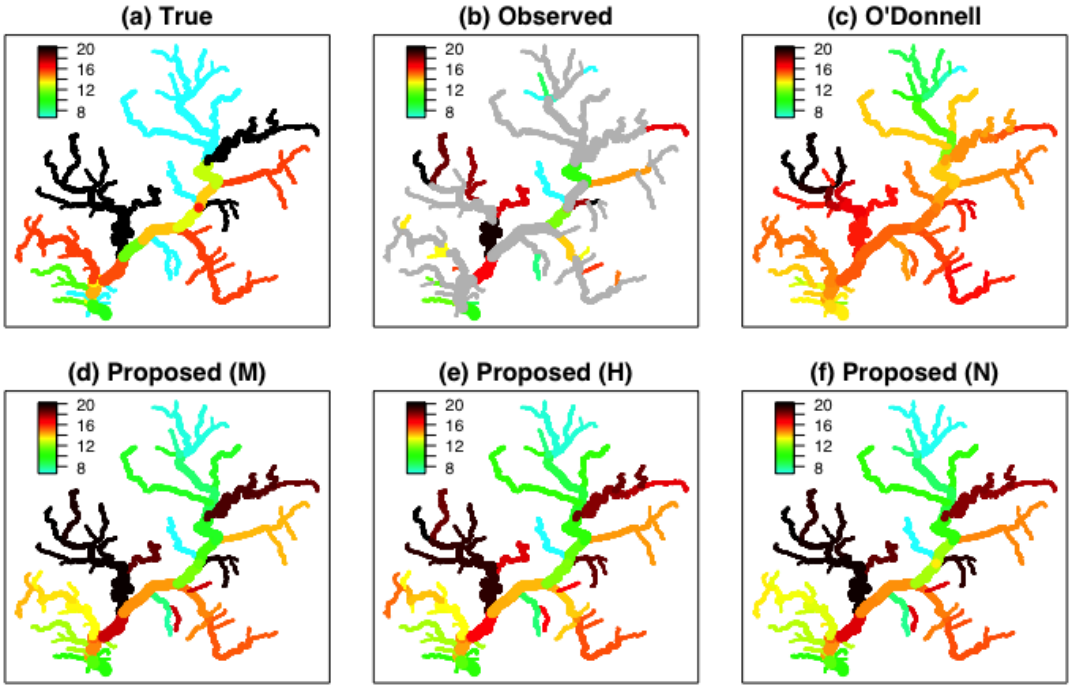

Figure 6: (a) True signal, (b) noisy observations when Obs=40 with unobserved segments shown in gray lines, (c) fit by O'Donnell *et al.* (2014), (d)-(e) fits by the proposed method with median thresholding and hard thresholding, and (f) fit by the proposed nondecimated method with median thresholding.

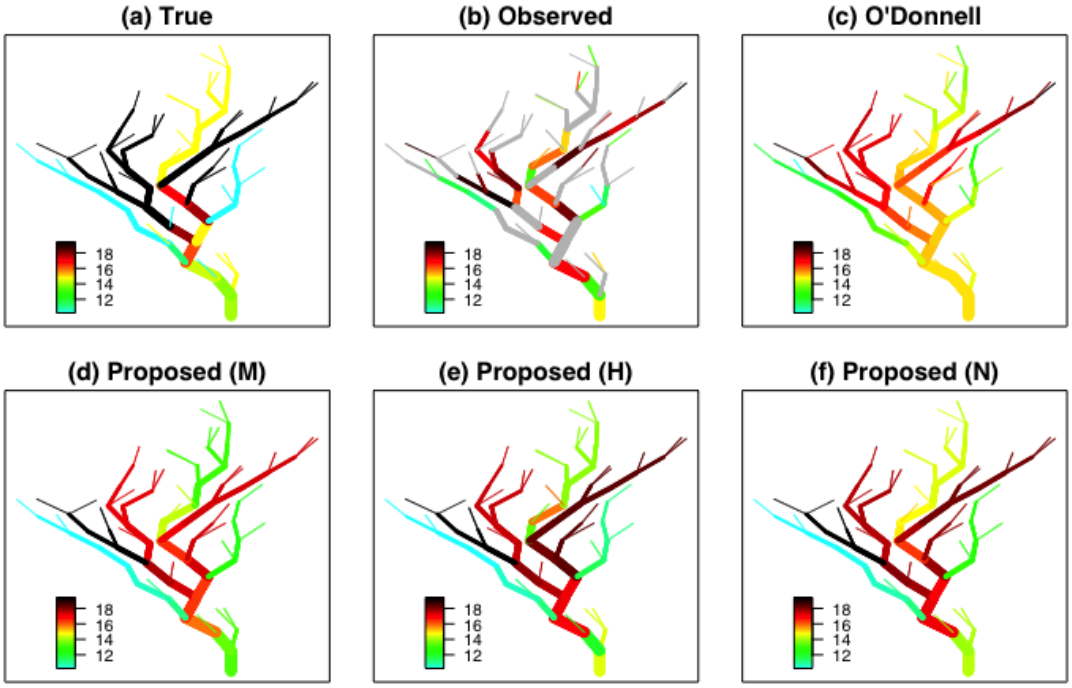

Figure 7: (a) True signal, (b) noisy observations when Obs=60 with unobserved segments shown in gray lines, (c) fit by O'Donnell *et al.* (2014), (d)-(e) fits by the proposed method with median thresholding and hard thresholding, and (f) fit by the proposed nondecimated method with median thresholding.

Table 3: Averages of RMSE values of streamflow data used in Gallacher *et al.* (2017) (standard errors) over 100 simulations.

RMSE (Std. error)	Obs=40			Obs=80		
	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
O'Donnell	1.7266 (0.2638)	1.8695 (0.2835)	1.9698 (0.2839)	1.2698 (0.1251)	1.3815 (0.1727)	1.5421 (0.2017)
Proposed (Median)	1.1716 (0.2530)	1.5464 (0.3611)	1.8121 (0.3539)	0.7265 (0.1212)	1.0249 (0.1510)	1.2818 (0.2599)
Proposed (Hard)	**1.1417 (0.2244)**	1.5769 (0.3050)	1.9040 (0.3286)	0.7396 (0.1317)	1.0666 (0.1678)	1.3705 (0.2495)
Proposed (Median, nlt)	1.1443 (0.2494)	**1.4988 (0.3267)**	**1.7329 (0.3385)**	**0.7162 (0.1219)**	**1.0106 (0.1678)**	**1.2816 (0.2712)**

6 Real data analysis

In this section, we apply the proposed lifting scheme to the real data set in Section 3. We consider the TOC water pollutant observed from 2012 to 2017. Water pollutants typically have some extreme values, which results in skewed empirical distributions. Therefore, we consider the average values of the log transformation of TOC data from 2012 to 2017 at each station shown in Figure 1. For the configuration of the results, we use an interpolation method based on equation (6). We consider the river network in Figure 4. Suppose that there are no observations in segment C. That is, we assume that the actual value of y_c is unknown. Then we interpolate the value of y_c with observations y_A, y_B, and y_E, which results in $\hat{y}_C = w_A y_A + w_B y_B + w_E y_E$, where $w_A + w_B + w_E = 1$. For the nondecimated version of the streamflow lifting scheme, clusters should be set up to achieve stable smoothing results. In this analysis, we only consider a permutation of observations in the same stream segments, assuming that the original removal path defined through Section 4 is such a well-behaved removal order. Twelve of the 127 stations are located in segments with two or more stations.

Assume that the underlying model of TOC data follows the model of (3). Although the actual function g is unknown, the similarity of interpolation can be evaluated by the approximated root mean square error ($\widetilde{\text{RMSE}}$),

$$\widetilde{\text{RMSE}} = \sqrt{\frac{\sum_{i=1}^{N_{tot}} (\tilde{g}(x_i) - \hat{g}(x_i))^2}{N_{tot}}},$$

where $N_{tot} = 942$ is the total number of stream segments in the Geum-River network, $\tilde{g}(x_i)$ denote the interpolation of raw data, and $\hat{g}(x_i)$ represents the interpolation of estimates.

Figure 8 shows the results by the proposed streamflow lifting scheme for the Miho-Cheon TOC data set. The interpolation for raw data is shown in panel (a). For comparison, we consider the method of O'Donnell *et al.* (2014). Its results are shown in panel (b). The interpolation results of the proposed methods are in panels (c) and (d), respectively. The nondecimated lifting scheme uses a bunch of random trajectories so that the results can vary over executions. From Figure 8, we observe that the high TOC values of the Geum-River downstream are affected by the water quality of the Miho-Cheon. In other words, TOC from the Miho-Cheon dominates the water pollution in the Geum-River downstream. As for Figure 3, we find that many industrial factories are near the Miho-Cheon catchment area. It

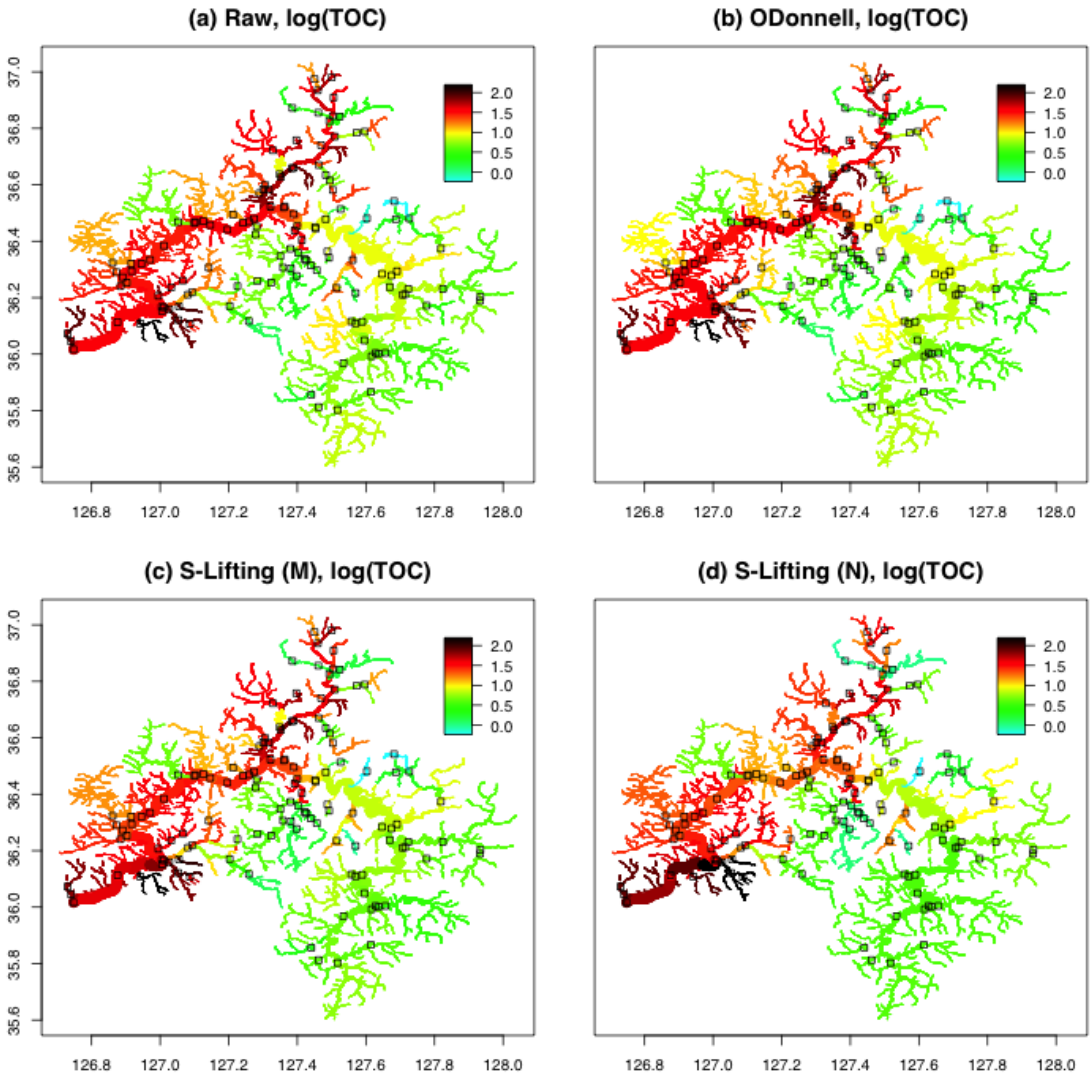

Figure 8: Real data analysis for TOC data. (a) Interpolation of row dataset, (b) interpolation of estimates by O'Donnell's method, (c) interpolation of estimates by the proposed streamflow lifting scheme with median thresholding, and (d) interpolation of estimates by the proposed nondecimated streamflow lifting scheme with median thresholding.

Table 4: $\widetilde{\text{RMSE}}$ results of the interpolation of the Geum-River data set.

	ODonnell	S-Lifting (M)	S-Lifting (N)
$\widetilde{\text{RMSE}}$	0.1240	0.0818	0.1856

is a plausible conclusion that industrial factories can affect the amount of TOC in the river network.

In addition, from Figure 8, all methods present smoothed results compared to the representation of the raw data. The proposed streamflow lifting schemes in panels (c) and (d) provide more smoothed estimation results. From panels (b) and (c), we observe that the O'Donnell's approach yields some orange colors on the Eastside while these colors are changed to green colors in the proposed streamflow lifting scheme. Moreover, the proposed lifting scheme decides some yellow values of the Westside on the network are noises. Therefore, the method also changes it into orange or red.

For the result of the nondecimated version, we find that some values on the east side of the network are high. The real data seems to be a mixed pattern instead of a piecewise function. So, it is challenging to establish appropriate clusters to carry out the nondecimated streamflow lifting scheme. Compared to this result, the original streamflow lifting scheme and the approach of O'Donnell *et al.* (2014) give more robust results. This observation is supported by the results of $\widetilde{\text{RMSE}}$ values listed in Table 4.

Before closing this section, we perform a multiscale analysis of streamflow data, which is one of the advantages of the proposed streamflow lifting scheme method. Let $g_4(x)$ be a representation at the finest level. We then decompose the function $g_4(x)$ into global component $g_3(x)$ and detailed component $d_3(x)$. It further breaks down the function $g_3(x)$ into global component $g_2(x)$ and detailed component $d_1(x)$. By repeating the above steps until a certain level 1, we finally decompose the $g_4(x)$ as

$$g_4(x) = g_1(x) + \sum_{\ell=1}^{3} d_\ell(x),$$

where ℓ denotes the resolution index. As ℓ decreases, the corresponding representation becomes coarser. To perform this multiscale analysis of the TOC data in the Geum-River network, we consider the representation in Figure 8(c) as the finest level representation $g_4(x)$.

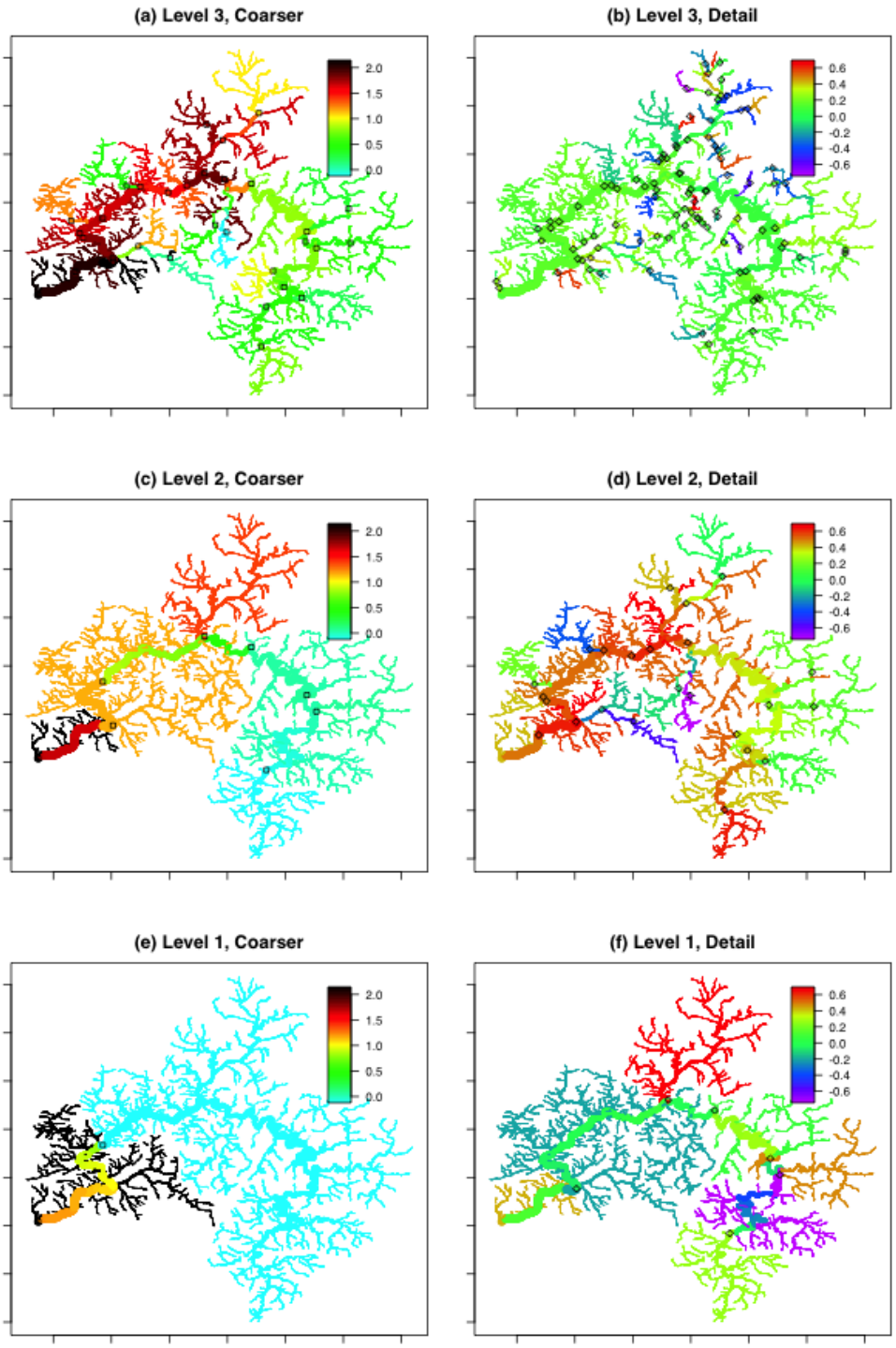

Figure 9: A multiscale analysis of TOC data. (a), (c) and (e) show global components of streamflow data at three different levels. (b), (d) and (f) are corresponding detail components. The number of points at each level (`nkeep`) are `32, 8, 2,` respectively.

Figure 9 shows the multiscale representations by the proposed streamflow lifting scheme. In Figure 9(a), we reconstruct the river network field only using 32 stations out of 127 stations, which still holds global features of the representation in Figure 8(c). The difference between the two representations is shown in Figure 9(a) as a detailed field. Figures 9(c) and (e) show the global components of the representation in Figure 8(c) using 8 stations and 2 stations, respectively, and Figures 9(d) and (f) show the corresponding differences. As the number of data points for reconstruction decreases, the corresponding representations are becoming rougher with focusing on global patterns. Instead, detail fields at each level provide some important information about networks that global components cannot represent.

7 Concluding remarks

In this paper, we have proposed a new lifting scheme for streamflow data. The proposed methods enable lifting scheme to streamflow data by (i) adopting a stream network adaptive neighborhood selection, (ii) constructing a prediction filter with flow-adaptive weighted averages, and (iii) setting a removal order by defining neighborhood flows of each observation point. By using the proposed neighborhood selection method, we reduce the number of boundary points and predict the values of upstream streamflow points. Besides, we have developed a nondecimated version of the proposed streamflow lifting scheme as a generalization. Simulation studies show that the proposed method works better than the conventional smoothing approach for streamflow data in particular situations, especially if there are some discontinuities in the data.

However, the proposed approach has some limitations. First, it is assumed that the volume of the water flow is proportional to the length of the segments and the Shreve order. In practice, however, the volume of the water flow may differ from the segment length and the Shreve order. The volume of the water varies over seasons. For example, precipitation in Korea is mostly is concentrated in the summer season. Second, in the simulation study, we have gathered segments in the given river network into several artificial groups to enhance the performance of the proposed lifting scheme. However, it is not easy to define optimal clusters in real data analysis. Therefore, one of the future studies will be to suggest an appropriate way to find optimal groups. Third, the removal order of the proposed method is

not determined by the value of the streamflow data set, but based on location only. If possible, a data-adaptive removal order selection algorithm is useful to enhance the performance of the proposed method.

Finally, the approach proposed in this study does not provide spatio-temporal data analysis. Since TOC data are observed irregularly in both space and time domains, it is necessary to have a novel method to carry out spatio-temporal streamflow data analysis. Lindström *et al.* (2014) and O'Donnell *et al.* (2014) solved this problem by calculating biweekly or monthly average data for each station. The method of O'Donnell *et al.* (2014) can then be used to build space-time basis functions with tensor products. However, if we find a way to construct multiscale spatio-temporal bases without merging the data, it will be more useful to capture the multiscale spatio-temporal behavior of the data. It is reserved for future research.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea government (2018R1D1A1B07042933). The author Seoncheol Park would like to acknowledge that this research was undertaken thanks in part to funding from the Canada First Research Excellence Fund (Global Water Futures: Solutions to Water Threats in an Era of Global Change, Climate-Related Precipitation Extremes project), the Pacific Climate Impacts Consortium and the Canadian Statistical Sciences Institute.

References

Artiola, J, Pepper, I. L., and Brusseau, M. L. (2004). *Environmental Monitoring and Characterization*. Elsevier Science and Technology Books.

Cressie, N., Frey, J., Harch, B., and Smith, M. (2006). Spatial prediction on a river network. *J. Agric. Biol. Environ. Stat.*, **11**, 127–150.

Gallacher, K., Miller, C., Scott, E. M., Willows, R., Pope, L., and Douglass, J. (2017). Flow-directed PCA for monitoring networks. *Environmetrics*, **28**, e2434.

Jansen, M. H., and Oonincx, P. (2005). *Second Generation Wavelets and Applications*. Springer Science and Business Media.

Jansen, M., Nason, G. P., and Silverman, B. W. (2009). Multiscale methods for data on graphs and irregular multidimensional situations. *J. Roy. Stat. Soc. B.*, **71**, 97–125.

Knight, M. I., and Nason, G. P. (2009). A 'nondecimated' lifting transform. *Stat Comput.*, **19**, 1–16.

Lindström, J., Szpiro, A. A., Sampson, P. D., Oron, A. P., Richards, M., Larson, T. V., and Sheppard, L. (2014). A flexible spatio-temporal model for air pollution with spatial and spatio-temporal covariates. *Environ. Ecol. Stat.*, **21**, 411–433.

Nunes, M. A., Knight, M. I., and Nason, G. P. (2006). Adaptive lifting for nonparametric regression. *Stat. Comput.*, **16**, 143–159.

O'Donnell, D., Rushworth, A., Bowman, A. W., Scott, E. M., and Hallard, M. (2014). Flexible regression models over river networks. *J. Roy. Stat. Soc. C.*, **63**, 47–63.

Sweldens, W. (1996). The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, **3**, 186–200.

Sweldens, W. (1998). The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, **29**, 511–546.

Ver Hoef, J. M. and Peterson, E. E., and Theobald, D. (2006). Spatial statistical models that use flow and stream distance. *Environ. Ecol. Stat.*, **13**, 449–464.

Ver Hoef, J. M., and Peterson, E. E. (2010). A moving average approach for spatial statistical models of stream networks. *J. Am. Stat. Assoc.*, **105**, 6–18.