Disaggregating Time-Series with Many Indicators: An Overview of the DisaggregateTS Package

by Luke Mosley, Kaveh Salehzadeh Nobari, Giuseppe Brandi, and Alex Gibberd

Abstract Low-frequency time-series (e.g., quarterly data) are often treated as benchmarks for interpolating to higher frequencies, since they generally exhibit greater precision and accuracy in contrast to their high-frequency counterparts (e.g., monthly data) reported by governmental bodies. An array of regression-based methods have been proposed in the literature which aim to estimate a target high-frequency series using higher frequency indicators. However, in the era of big data and with the prevalence of large volumes of administrative data-sources there is a need to extend traditional methods to work in high-dimensional settings, i.e., where the number of indicators is similar or larger than the number of low-frequency samples. The package DisaggregateTS includes both classical regressions-based disaggregation methods alongside recent extensions to high-dimensional settings. This paper provides guidance on how to implement these methods via the package in R, and demonstrates their use in an application to disaggregating CO2 emissions.

1 Introduction

Economic and administrative data, such as recorded surveys and consensus, are often disseminated by international governmental agencies at low or inconsistent frequencies, or irregularly-spaced intervals. To aid the forecasting of the evolution of the dynamics of these macroeconomic and socioeconomic indicators, as well as their comparison with higher resolution indicators provided by international agencies, statistical agencies rely on signal extraction, interpolation and temporal distribution adjustments of the low-frequency data to provide high precision and uninterrupted historical data. Although, temporal distribution, interpolation and benchmarking are closely associated with one another, this article and its respective package (**DisaggregateTS**, Mosley and S. Nobari, 2024), expend particular attention to interpolation and temporal distribution (disaggregation) techniques, where the latter is predicated on regression-based methods[1]. These regression-based temporal distribution techniques rely on high-frequency indicators to estimate (relatively) accurate high-frequency data points. With the prevalence of large volume of high-frequency administrative data, a great body of literature pertaining to statistical and machine learning methods has been dedicated to taking advantage of these additional resources for forecasting purposes (see Fuleky, 2019, for an overview of macroeconomic forecasting in the presence of big data). Additionally, one may wish to utilize these abundant indicators to generate high-frequency estimates of low-frequency time-series with greater precision. However, in high-dimensional linear regression models where the number of dimensions surpass that of the observations, consistent estimates of the parameters is not possible without imposing additional structure (see Wainwright, 2019). Hence, this article and the package **DisaggregateTS** adapt recent contributions in high-dimensional temporal disaggregation (see Mosley et al., 2022) to extend previous work within this domain (see the package **tempdisagg** Sax et al., 2023, and its corresponding article Sax and Steiner (2013)) to high-dimensional settings.

As noted by Dagum and Cholette (2006), time-series data reported by most governmental and administrative agencies tend to be of low-frequency and precise, but not particularly timely, whereas their high-frequency counterparts seldom uphold the same degree of precision.

The aim of temporal distribution techniques is to generate high-frequency estimates that can track shorter term movements, than directly observable with the direct low-frequency observations. While interpolation problems are generally encountered in the context of stock series, where say, the quarterly value of the low-frequency series must coincide with the value of third month of the high-frequency data (of the same quarter), temporal distribution problems often concern flow series, where instead the value of the low-frequency quarterly series must agree with the sum (or weighted combination) of the values of the high-frequency months in that quarter. The latter approach is generally accomplished by identifying and taking advantage of a number of high-frequency indicators which are deemed to behave in a similar manner to the low-frequency series, and by estimating the high-frequency series through a linear combination of such indicators.

In the last few decades, a significant number of articles have been published within this domain—

[1] See Dagum and Cholette (2006) for an overview of benchmarking, interpolation, temporal distribution and calendarization techniques.

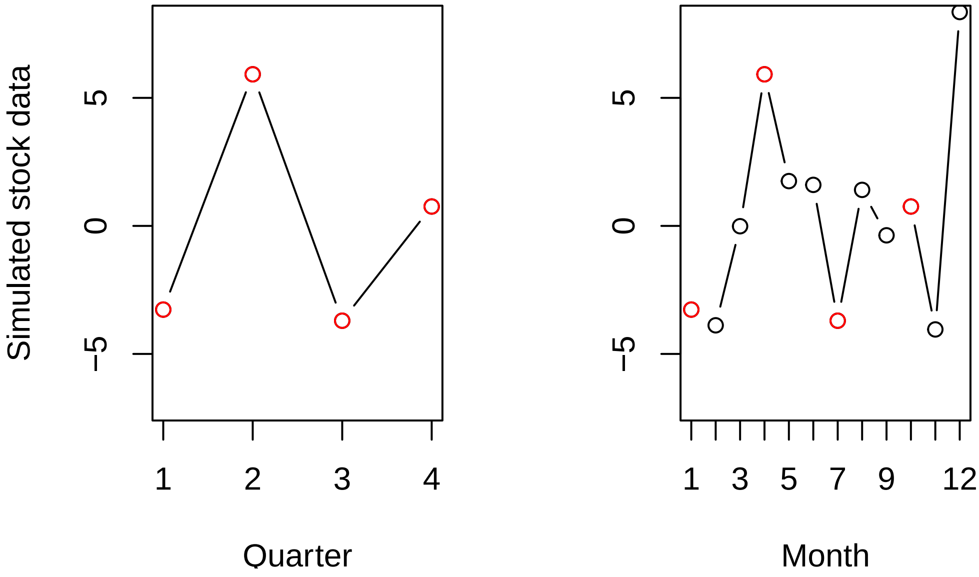

Figure 1: Quarterly and monthly simulated stock data

see Dagum and Cholette (2006) for a detailed review of these techniques. Notable studies within this context include the additive regression-based benchmarking methods of Denton (1971) and Chow and Lin (1971), Chow and Lin (1976), as well as those proposed by Fernández (1981) and Litterman (1983) in the presence of highly persistent error processes. More recently these methods have been extended to the high-dimensional setting by Mosley et al. (2022), where prior information on the structure of the linear regression model is used to enable estimation, and better condition the regression problem. Specifically, this is accomplished by "least absolute shrinkage and selection operator" (LASSO hereafter) proposed by Tibshirani (1996), which in principle selects an appropriate model by penalizing the coefficients (in scale) of the high-dimensional regression, in effect discarding the irrelevant indicators from the model. In what follows, we demonstrate how to apply these methods using **DisaggregateTS** to easily estimate high-frequency series of interest.

The remainder of the paper is organized as follows: Section 2 presents the methodologies behind key temporal disaggregation techniques included in the **DisaggregateTS** package, along with their extensions to high-dimensional settings. Section 3 introduces the **DisaggregateTS** package and highlights its key functions. Sections 4 and 5 provide examples based on simulations (using a function in the package that generates synthetic data) and empirical data to demonstrate the package's functionality. Finally, Section 6 concludes the paper.

2 Sparse temporal disaggregation

2.1 Classical regression-based techniques

The data in figures 1 and 2 are generated using the `TempDisaggDGP()` function from the **DisaggregateTS** package, representing simulated stock and flow data. For the simulated stock data (Figure 1, each quarter's low-frequency data should match the first month's value of the corresponding high-frequency series, denoted with red dots. For the simulated flow data (Figure 2, the quarterly figures should equal the sum of the sub-quarterly values

Suppose we observe a low-frequency series, say, quarterly GDP, encoded as the vector $\mathbf{y}_q \in \mathbb{R}^n$, containing n quarterly observations. We desire to disaggregate this series to higher frequencies (say monthly), where the disaggregated series is denoted $\mathbf{y}_m \in \mathbb{R}^p$, with $p = 3n$. Furthermore, we wish that the disaggregated series be temporally consistent without exhibiting any jumps between quarters (see Section 3.4 of Dagum and Cholette, 2006, for examples of such inconsistencies between the periods). The challenge is to identify an approach that distributes the variation between each observed quarterly point to the monthly level. A method that has been extensively studied in the literature concerns finding high-frequency (e.g., monthly) indicator series that are thought to exhibit similar inter-quarterly movements as the low-frequency variable of interest. Let us denote a set of p observations from these d indicators as the matrix $\mathbf{X}_m \in \mathbb{R}^{p\times d}$. A classical approach to provide high-frequency estimates is the regression-based temporal disaggregation technique proposed by Chow and Lin (1971) whereby the unobserved monthly series $\mathbf{y}_m$ are assumed to follow the regression:

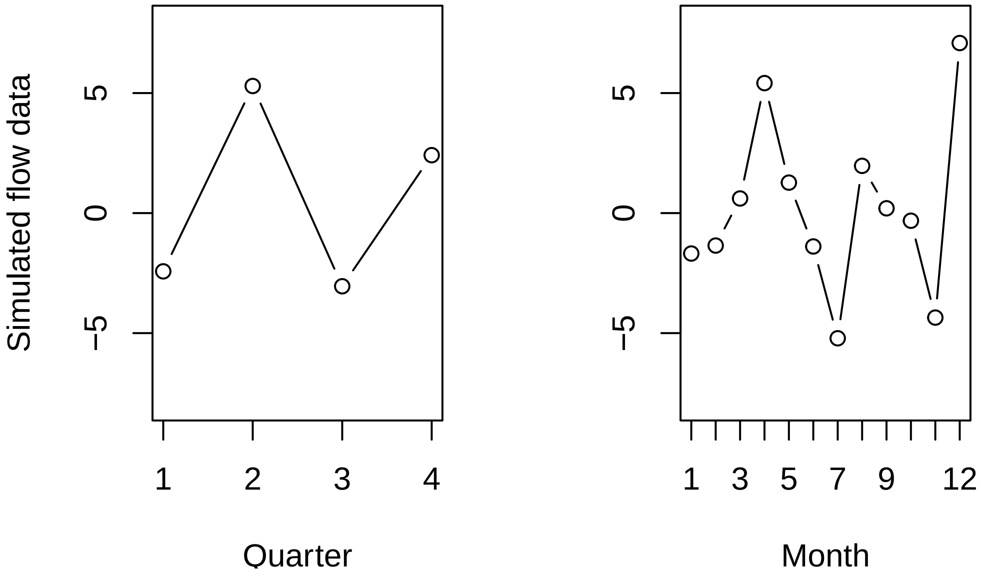

Figure 2: Quarterly and monthly simulated flow data

$$\mathbf{y}_m = \mathbf{X}_m\beta + \mathbf{u}_m, \quad \mathbf{u}_m \sim N(\mathbf{0}_m, \mathbf{V}_m) \tag{1}$$

where $\beta \in \mathbb{R}^d$ is a vector of regression coefficients to be estimated (noting that $\mathbf{X}_m$ may contain deterministic terms) and $\mathbf{u}_m \in \mathbb{R}^p$ is a vector of residuals. Chow and Lin (1971) assume that $\mathbf{u}_m$ follows as AR(1) process of the form $u_t = \rho u_{t-1} + \varepsilon_t$ with $\varepsilon_t \sim N(0, \sigma^2)$ and $|\rho| < 1$. The assumption of stationary residuals allows for a cointegrating relationship between $\mathbf{y}_m$ and $\mathbf{X}_m$ when they are integrated of the same order. Thus, the covariance matrix has a well-known Toeplitz structure as follows:

$$\mathbf{V}_m = \frac{\sigma^2}{1-\rho^2}\begin{pmatrix} 1 & \rho & \cdots & \rho^{p-1} \\ \rho & 1 & \cdots & \rho^{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{p-1} & \rho^{p-2} & \cdots & 1 \end{pmatrix} \tag{2}$$

where ρ and σ are unknown parameters that need to be estimated. The dependent variable $\mathbf{y}_m$ in (1) is unobserved, hence the regression is premultiplied by the $n \times p$ aggregation matrix $\mathbf{C}$, where:

$$\begin{aligned} \mathbf{C} &= \mathbf{I}_n \otimes (1,1,1) \\ &= \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}_{n\times p} \end{aligned} \tag{3}$$

where $\otimes$ is the Kronecker operator, and the vector of ones in (3) is used for flow data (e.g., GDP), such that the sum of the monthly GDPs coincides with its quarterly counterpart[2]. The premultiplication yields the quarterly counterpart of (1):

$$\mathbf{C}\mathbf{y}_m = \mathbf{C}\mathbf{X}_m\beta + \mathbf{C}\mathbf{u}_m, \quad \mathbf{C}\mathbf{u}_m \sim N(\mathbf{C}\mathbf{0}_m, \mathbf{C}\mathbf{V}_m\mathbf{C}^\top). \tag{4}$$

The GLS estimator for β is thus expressed as follows:

$$\hat{\beta} = \arg\min_{\beta\in\mathbb{R}^d}\left\{\left\|\mathbf{V}_q^{-\frac{1}{2}}(\mathbf{y}_q - \mathbf{X}_q\beta)\right|_2^2\right\} \tag{5}$$

$$= \left(\mathbf{X}_q^\top\mathbf{V}_q^{-1}\mathbf{X}_q\right)^{-1}\mathbf{X}_q^\top\mathbf{V}_q^{-1}\mathbf{y}_q \tag{6}$$

where $\mathbf{X}_q = \mathbf{C}\mathbf{X}_m$, $\mathbf{y}_q = \mathbf{C}\mathbf{y}_m$ and $\mathbf{V}_q = \mathbf{C}\mathbf{V}_m\mathbf{C}^\top$. Note that estimating β requires the knowledge of the unknown parameters σ and ρ in $\mathbf{V}_m$ which are unknown. We employ the profile-likelihood maximization technique of Bournay and Laroque (1979), which involves first estimating $\hat{\beta}$ and $\mathbf{V}_q$

[2] For alternative aggregations see Quilis (2018) and Sax et al. (2023). For instance, if quarterly values correspond to averages of monthly values, then the vector in equation (3) assumes the form $(0.33, 0.33, 0.33)$.

conditional on a given value of ρ, and maximizing the log-likelihood function by conducting a grid search over $\rho \in (-1, 1)$ for the autoregressive parameter. However, in practical applications, including well-known implementations such as the **tempdisagg** package and the `Gretl` econometric software Cottrell and Lucchetti (2023), the grid search is often restricted to $\rho \in [0, 1)$, reflecting a non-negative constraint on the autoregressive parameter. Our method is specifically designed to search within $\rho \in [0, 1)$, reflecting this non-negative constraint.

Chow and Lin (1971) show the optimal solution is obtained by:

$$\hat{\mathbf{y}}_m = \mathbf{X}_m \hat{\beta} + \hat{\mathbf{V}}_m \mathbf{C} \hat{\mathbf{V}}_q^{-1} \left(\mathbf{y}_q - \mathbf{X}_q \hat{\beta} \right), \tag{7}$$

where $\mathbf{X}_m \hat{\beta}$ is the conditional expectation of $\mathbf{y}_m$ given $\mathbf{X}_m$ and the estimate of the monthly residuals are obtained by disaggregating the quarterly residuals $\mathbf{y}_q - \mathbf{X}_q \hat{\beta}$ to attain temporal consistency between $\hat{\mathbf{y}}_m$ and $\mathbf{y}$.

Other variants of the regression-based techniques include those proposed by Denton (1971), Fernández (1981), Litterman (1983), and Cholette (1984), with the latter two addressing scenarios where $\mathbf{y}_m$ and $\mathbf{X}_m$ are not cointegrated. Although these traditional techniques are included in the **DisaggregateTS** package, a comprehensive overview of different temporal disaggregation techniques and distribution matrices can be found in Table 2 of Sax and Steiner (2013). The **tempdisagg** package implements several standard methods for temporal disaggregation, each with distinct approaches for estimating high-frequency series. These include the Denton, Denton-Cholette, Chow-Lin, fernández, and Litterman methods. As summarized in Table 2 of Sax and Steiner (2013), Denton and Denton-Cholette focus on movement preservation, while regression-based methods like Chow-Lin, fernández, and Litterman perform generalized least squares regressions on the low-frequency series using one or more high-frequency indicators. We have implemented the Chow-Lin method with a non-negative autoregressive parameter, constrained within the range $[0, 1)$, following the approach of Bournay and Laroque (1979) and Cottrell and Lucchetti (2023). This offers flexibility in different disaggregation scenarios.

2.2 Extension to high-dimensional settings

The shortcoming of Chow and Lin (1971) becomes evident in data-rich environments, where the number of indicators $d \gg n$ surpass that of the time-stamps for the low-frequency data. Let us once again recall the GLS estimator (6). When $d < n$ and the columns of $\mathbf{X}_q^\top \mathbf{V}_q^{-1} \mathbf{X}_q$ are independent, the estimator is well-defined. However, when $d > n$, the matrix is rank-deficient - i.e., $\operatorname{rank}(\mathbf{X}_q^\top \mathbf{V}_q^{-1} \mathbf{X}_q) \le \min(n, d)$, the matrix $\mathbf{X}_q^\top \mathbf{V}_q^{-1} \mathbf{X}_q$ has linearly dependent columns, and thus is not invertible. In moderate dimensions, where $d \approx n$, $\mathbf{X}_q^\top \mathbf{V}_q^{-1} \mathbf{X}_q$ has eigenvalues close to zero, leading to high variance estimates of $\hat{\beta}$.

Mosley et al. (2022) resolve this problem by adding a regularizing penalty (e.g., ℓ_1 regularizer) onto the GLS cost function (5):

$$\hat{\beta}(\lambda_n \mid \rho) = \arg\min_{\beta \in \mathbb{R}^d} \left\{ \left\| \mathbf{V}_q^{-\frac{1}{2}} (\mathbf{y}_q - \mathbf{X}_q \beta) \right\|_2^2 + \lambda_n \|\beta\|_1 \right\}. \tag{8}$$

Unlike the GLS estimator (6), the regularized estimator corresponding to the cost function (8) is a function of λ_n and the autoregressive parameter ρ. Henceforth, it is important to nominate the most suitable λ_n and ρ to correctly recover the parameters. In (8), we denote the estimator as $\hat{\beta}(\lambda_n \mid \rho)$ to highlight that the solution paths of the estimator for different values of λ_n, say, $\lambda_n^{(1)}, \lambda_n^{(2)}, \cdots, \lambda_n^{(k)}$ are generated for (i.e. conditional on) a fixed ρ. The solution paths are obtained using the LARS algorithm proposed by Efron et al. (2004), the benefits of which have been extensively discussed in Mosley et al. (2022).

LASSO estimators inherently exhibit a small bias, such that $\|\hat{\beta}\|_2^2 \le \|\beta^*\|_2^2$, where β^* denotes the true coefficient vector. To alleviate this issue, Mosley et al. (2022) further follow Belloni and Chernozhukov (2013), by performing a refitting procedure using least squares re-estimation. The latter entails generating a new $n \times d^{(l)}$ sub-matrix $\mathbf{X'}_q$, where $d^{(l)} \le d$ from the original $n \times d$ matrix $\mathbf{X}_q$, with $\mathbf{X'}_q$ corresponding to the columns of $\mathbf{X}_q$ supported by $\hat{\beta}(\lambda_n^{(l)} \mid \rho)$, for solutions $l = 1, \cdots, k$ obtained from the LARS algorithm[3]. We then perform a usual least squares estimation on $(\mathbf{y}_q, \mathbf{X'}_q)$ to obtain debiased solution paths for each $\lambda_n^{(l)}$.

Finally, Mosley et al. (2022) choose the optimal estimate from $\hat{\beta}(\lambda_n^1 \mid \rho), \cdots, \hat{\beta}(\lambda_n^k \mid \rho)$ using the Bayesian Information Criterion (BIC hereafter) proposed by Schwarz (1978). The motivation for nomi-

[3] noting the LARS algorithm produces solutions evaluated at a series of $\{\lambda_l\}_{l=1}^k$ points.

nating this statistic over resampling methods, such as cross-validation or bootstrapping techniques, stems from the small sample size in the low-frequency observations. The optimal regularization is chosen conditional on ρ according to

$$\hat{\lambda}_n(\rho) = \arg\min_{\lambda_n(\rho)\in\{\lambda_n^{(1)}(\rho),\cdots,\lambda_n^{(k)}(\rho)\}} \left\{-2\mathcal{L}\left(\hat{\beta}(\lambda_n \mid \rho), \hat{\sigma}^2\right) + \log(n)K_{\lambda_n(\rho)}\right\}, \tag{9}$$

where $K_{\lambda_n(\rho)} = |\{r : (\hat{\beta}(\lambda_n \mid \rho)_r \neq 0)\}|$ is the degrees of freedom and $\mathcal{L}(\hat{\beta}(\lambda_n \mid \rho), \hat{\sigma}^2)$ is the log-likelihood function of the GLS regression (6), which in the presence of Gaussian errors, is given by:

$$\mathcal{L}(\hat{\beta}, \hat{\sigma}^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log(|\mathbf{S}|) - \frac{1}{2\sigma^2}(\mathbf{y}_q - \mathbf{X}_q\beta)^\top(\mathbf{y}_q - \mathbf{X}_q\beta), \tag{10}$$

where $|\mathbf{S}|$ is the determinant of the Toeplitz matrix $\mathbf{S}$ depending on ρ, such that $\mathbf{V}_q = \sigma^2\mathbf{S}$ (recalling that $\mathbf{V}_q = \mathbf{C}\mathbf{V}_m\mathbf{C}^\top$).

3 The package

In this Section, we showcase the main functions that has been included in the **DisaggregateTS** package. Following Sax et al. (2023), we first introduce the main function of the package and it its features, and subsequently we will showcase other functions that allow the practitioner to conducting simulations and analyses.

3.1 Functions

The main function of the package which performs the sparse temporal disaggregation method proposed by Mosley et al. (2022) is `disaggregate()`. This function is of the following form:

```
disaggregate(Y, X, aggMat, aggRatio, method, ...)
```

where the first argument of the function, `Y`, corresponds to the $n \times 1$ vector of low-frequency series $\mathbf{y}_q$ that we wish to disaggregate, and the second argument, `X`, is the $p \times d$ matrix of high-frequency indicator series $\mathbf{X}_m$. In the event that there is no input `X`, the disaggregation matrix $\mathbf{X}_m$ is replaced with an $n \times 1$ vector of ones.

The argument `aggMat` coincides with the aggregation matrix $\mathbf{C}$ in (3), and it has been set to `"sum"` by default, rendering it suitable for flow data. Alternative options include `"first"`, `"last"` and `"average"`. The aggregation (distribution) matrices that are utilized in this function are summarized in table 2 of Sax et al. (2023).

The argument `aggRatio` has been set to `4` by default, which represents the ratio of annual to quarterly data. In general, this argument should be set to the ratio of the high-frequency to low-frequency series. For instance, in the examples considered in the preceding Sections, we had considered quarterly data as the low-frequency series, and monthly data as its high-frequency counterpart. Thus, in this setting `aggRatio = 3`. At first glance, the presence of the `aggRatio` argument may seem redundant. However, if $n \geq n_l \times \text{aggRatio}$, then extrapolation is done up to n.

Finally, the argument `method` refers to the method of disaggregation under consideration. This argument has been set to `"Chow-lin"` method by default, which is the classical regression-based disaggregation technique introduced in Section 2.1. Other classical low-dimensional options include `"Denton"`, `"Denton-Cholette"`, `"fernandez"`, and `"Litterman"`, where these techniques have been extensively discussed in Dagum and Cholette (2006) and Sax and Steiner (2013). The main contribution of this package stems from the `"spTD"` and `"adaptive-spTD"` options pertaining to sparse temporal disaggregation and adaptive sparse temporal disaggregation, which are Mosley et al. (2022)'s high-dimensional extension of the regression-based techniques proposed by Chow and Lin (1971). In a high-dimensional regression, the adaptive LASSO is relevant when, for instance, the columns of the design matrix $\mathbf{X}$ exhibit multicollinearity, and the *Irrepresentability Condition* (IC hereafter) is violated (see Zou, 2006, for details). In such settings, the regularization parameter λ does not satisfy the oracle property, which can lead to inconsistent variable selection. The adaptive counterpart of the regularized GLS cost function (8), can be expressed as follows:

$$\hat{\beta}(\lambda_n \mid \rho) = \arg\min_{\beta\in\mathbb{R}^d}\left\{\left\|\mathbf{V}_q^{-\frac{1}{2}}(\mathbf{y}_q - \mathbf{X}_q\beta)\right\|_2^2 + \lambda_n\sum_{j=1}^{d}\frac{|\beta_j|}{|\hat{\beta}_{\text{init},j}|}\right\}, \tag{11}$$

where $\hat{\beta}_{\text{init},j}$ is an initial estimator, predicated on $\hat{\beta}(\hat{\rho})$ from the regularized (LASSO) temporal disaggregation. See Mosley et al. (2022), for the details of the proposed methodology, and Zou (2006) and

Van de Geer et al. (2011) to yield variable selection consistency using the OLS estimator and LASSO as $\hat{\beta}_{\text{init},j}$ when the IC condition is violated.

The second main function of the **DisaggregateTS** package is `TempDisaggDGP()`, which generates synthetic data that can be used for conducting simulations using the `disaggregate()` function. The main arguments of this function are as follows:

```
TempDisaggDGP(n_l, n, aggRatio, p, beta, sparsity, method, aggMat, rho, ...)
```

where the first argument corresponds to the size of low-frequency series and `n` to that of the high-frequency series. Moreover, `aggRatio` and `aggMat` are defined as before, in turn representing the ratio of the high-frequency to low-frequency series, as well as the aggregation matrix (3). A minor difference in the DGP function is that if $n \geq n_l \times \text{aggRatio}$, then the last $n - \text{aggRatio} \times n_l$ columns of the aggregation matrix are set to zero, such that Y is observed only up to n_l. Argument `p` sets the dimensionality of high-frequency series (set to 1 by default), `beta` which has been set to 1 by default is the positive and negative elements of the coefficient vector, `sparsity` is the sparsity percentage of the coefficient vector, and `rho` is the autocorrelation of the error terms, which has been set to `0` by default. Finally, the `method` argument determines the data generating process of the error terms, corresponding to methods discussed earlier in this Section.

A number of optional arguments included in the function determine the mean vector and the standard deviation of the design matrix, as well as options such a setting seed for running the simulations, where the design matrix and the coefficient vectors are fixed.

In what follows, we showcase a simple example of the function and its respective outputs:

```
# Generate low-frequency quarterly series and its high-frequency monthly counterpart
SynthethicData <- TempDisaggDGP(n_l = 2,
                                n = 6,
                                aggRatio = 3,
                                p = 6,
                                beta = 0.5,
                                sparsity = 0.5,
                                method  = 'Chow-Lin',
                                rho = 0.5)
```

In the example above, we generate low-frequency series $\mathbf{y}_q \in \mathbb{R}^2$ corresponding to two quarters, and consequently, its high-frequency monthly counterpart $\mathbf{y}_m \in \mathbb{R}^6$. It is further assumed that the data is generated using six monthly indicators - i.e., $\mathbf{X}_m^{6\times6}$, with a coefficient vector $\beta \in \mathbb{R}^6$, where $\beta_j \in \{-0.5, 0, +0.5\}$. Since, the sparsity argument is set to 0.5, only half of β's elements are non-zero. Finally, the error vector $\mathbf{u}_m$ is assumed to follow the AR(1) structure of Chow and Lin (1971), with an autocorrelation parameter of $\rho = 0.5$.

4 Simulations

In this Section, we show a simulation exercise to demonstrate the implementation of the temporal disaggregation method via the **DisaggregateTS** package.

4.1 Classical setting

We start by simulating the dependent variable $Y \in \mathbb{R}^{17}$ and the set of high-frequency exogenous variables $X \in \mathbb{R}^{68\times5}$ by using the command:

```
# Set seed for reproducibility
set.seed(27)
# Generate low-frequency yearly series and its high-frequency quarterly counterpart
n_l <- 17 # The number of low-frequency data points - annual
n <- 68 # The number of high-frequency (quarterly) data points.
p_sim <- 5 # The number of the high-frequency exogenous variables.
rho_sim <- 0.8 # autocorrelation parameter
Sim_data <- TempDisaggDGP(n_l,
                          n,
                          aggRatio = 4,
                          p = p_sim,
```

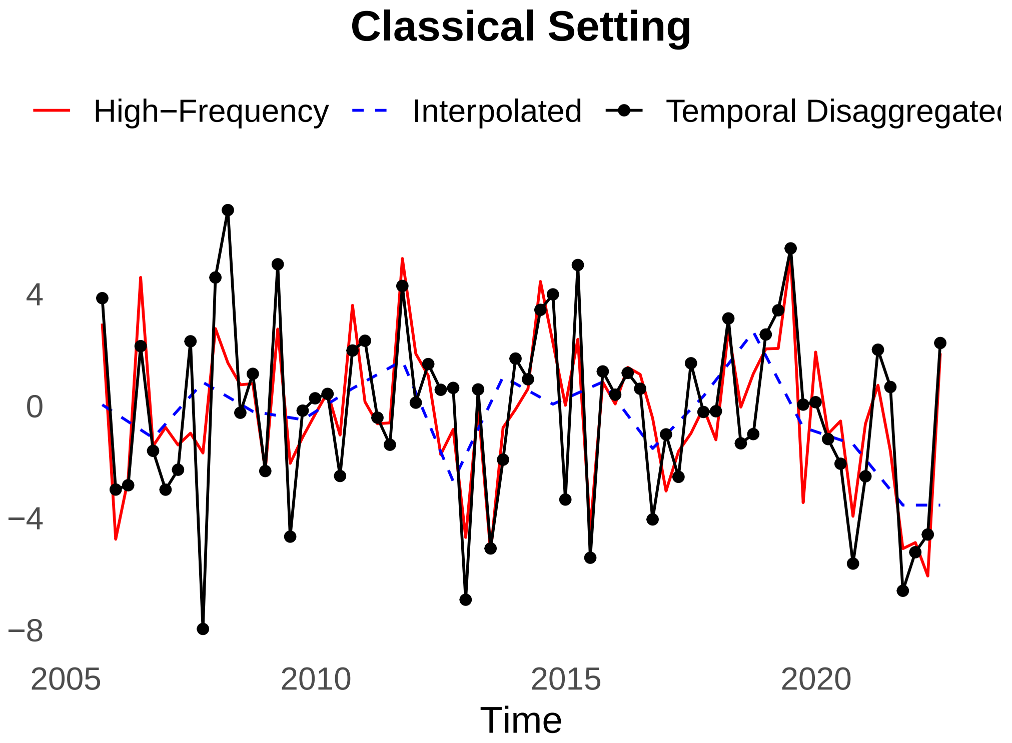

Figure 3: Temporal disaggregated and interpolated observations for the estimation under the classical setting. The plot is built using the snippet code provided in this subsection. As we used the setting `aggMat = sum`, the sum of every four disaggregated observations corresponds to an actual low-frequency observation.

```
                          rho = rho_sim)
Y_sim <- matrix(Sim_data$Y_Gen) # Extract the simulated dependent low-frequency variable
X_sim <- matrix(Sim_data$X_Gen) # Extract the simulated dependent
# (low-frequency) variable
Y_sim_HF_obs <- matrix(Sim_data$y_Gen) #  HF simulated observations
```

In this example, we are generating a set of low-frequency data, i.e. 17 annual datapoints and a set of high-frequency (quarterly) exogenous variables that we want to use to infer the high-frequency counterpart of the low-frequency data. We now want to temporally disaggregate the low-frequency time series by using the information encapsulated in the high-frequency time series. In this case, since the number of time observations is larger than the number of exogenous variables, we can use standard methodologies to estimate the temporal disaggregation model. To do so, we use the `disaggregate()` function setting `method="Chow-Lin"`. The code is as follows:

```
C_sparse_SIM <- disaggregate(Y_sim,
                             X_sim,
                             aggMat = "sum",
                             aggRatio = 4,
                             method = "Chow-Lin")
C_sparse_SIM$beta_Est

#> 1 x 1 Matrix of class "dgeMatrix"
#>           [,1]
#> [1,] 0.3193439

Y_HF_SIM <- C_sparse_SIM$y_Est[ ,1] # Extract the temporal disaggregated
# dependent variable estimated through the function disaggregate()
```

We show in Figure 3 the results, where we depict the high-frequency observation computed via standard interpolation and estimated through the Chow-Lin temporal disaggregation method.

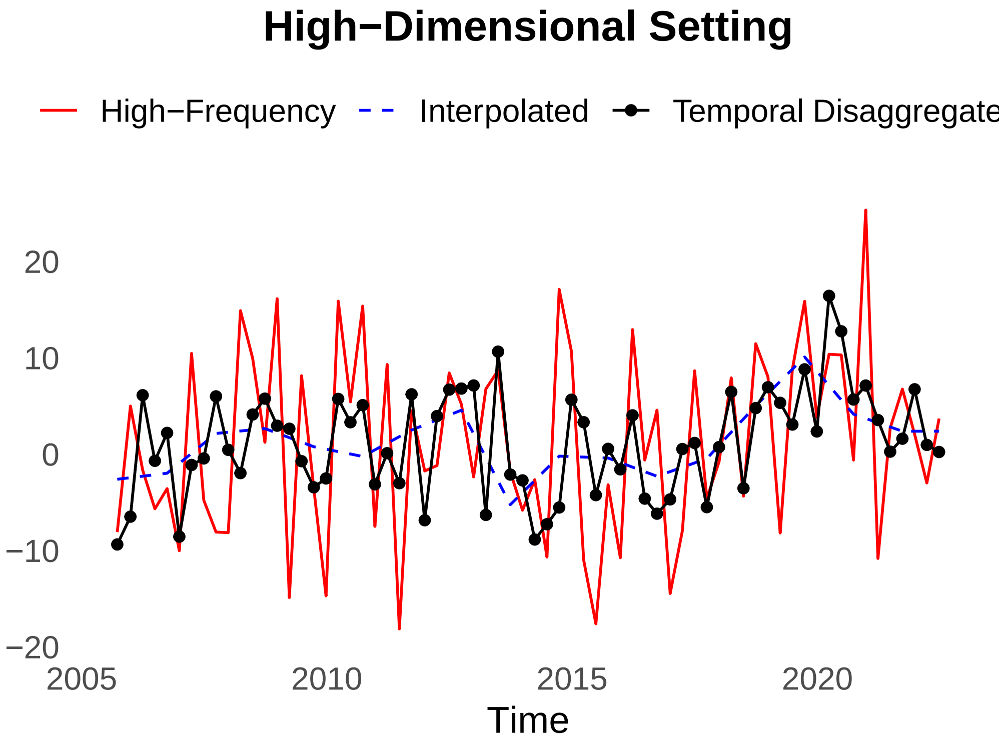

Figure 4: Temporal disaggregated and interpolated observations for the estimation under the high-dimensional setting. The plot is built using the snippet code provided in this subsection. As we used the setting `aggMat = sum`, the sum of every four disaggregated observations corresponds to an actual low-frequency observation.

4.2 High-dimensional setting

We now repeat the simulation experiment in a high-dimensional setting, where the number of temporal observations is lower than the number of exogenous variables. In this case, standard methods like Chow-Lin cannot be applied. To do so, we simulate the dependent variable $Y \in \mathbb{R}^{17}$ as before, but now the set of high-frequency exogenous variables is of dimension $X \in \mathbb{R}^{68\times 100}$. Similarly, as before, we can use the following command:

```
# Generate low-frequency yearly series and its high-frequency quarterly counterpart
set.seed(27)
n_l <- 17 # The number of low-frequency data points
n <- 68 # The number of high-frequency data points - quarterly
p_sim <- 100 # The number of the high-frequency exogenous variables.
rho_sim <- 0.8 # autocorrelation parameter
Sim_data <- TempDisaggDGP(n_l,
                          n,
                          aggRatio = 4,
                          p = p_sim,
                          rho = rho_sim)
Y_sim <- matrix(Sim_data$Y_Gen) #Extract the simulated dependent
# (low-frequency) variable
X_sim <- Sim_data$X_Gen #Extract the simulated exogenous variables - high-frequency
```

In this case, we cannot use a standard technique, and we need to estimate a sparse model to overcome the curse of dimensionality. The `disaggregate()` function can handle the high-dimensional setting by choosing the method to be `"spTD"` or `"adaptive-spTD"`. In the following example, we use the latter:

As we can see from both Figures 3 and 4, the standard interpolation cannot reproduce the fluctuations of the data, making the result overly smooth. On the other hand, the temporal disaggregation methods demonstrate fluctuations in line with the actual observations. We remark that the degree of success in recovering the short-term fluctuations depends considerably on the setting, and as one may expect, performance is not as strong in the high-dimensional scenario where the estimator must simultaneously search for appropriate indicators and estimate the parameters.

We would in general caution against throwing sets of non-curated (i.e. possibly irrelevant) indicators into any of the Chow-Lin methods. Instead, we recommend that where possible, indicators that are designed to track the outcome of interest are used, e.g., we may wish to benchmark (align) monthly flash-indicators of GDP against more reliable yearly/quarterly observations, in this case, the flash estimates can be used as an indicator within a standard Chow-Lin approach. If there is a larger set of indicators which practitioners find hard to decide amongst, that is all indicators could be feasible from the practitioner view, then we recommend users may try the model-selection based `"spTD"`, and `"adaptive-spTD"` routines. In some settings, use of the classical Chow-Lin procedure becomes impossible due to the lack of low-frequency responses, in such settings it is necessary to either apply regularization or perform model-selection.

To conclude the simulation exercise, we present a set of statistics from 1,000 simulations for both the linearly interpolated and temporally disaggregated time series across the two simulation settings. The results are summarized in Table 1. As it can be observed, the temporally disaggregated time series provides a much better representation of the real data's level of dispersion, while interpolation consistently underestimates it. In terms of MSE and MAE, the findings are mixed: for the classic case, temporal disaggregation shows lower values for both metrics, whereas in the high-dimensional setting, the opposite trend is observed.

Table 1: Mean and standard deviation (in parentheses) of key statistical measures for the high-frequency simulated observations, temporal disaggregated observations, and interpolated observations across 1000 Monte Carlo simulations. MAE and MSE are computed with respect to the high-frequency simulated observations.

Statistic	Classical_Obs	Classical_Temporal_Disaggregation	Classical_Interpolation	HighDim_Obs	HighDim_Temporal_Disaggregation	HighDim_Interpolation
Standard Deviation	2.716 (0.258)	2.888 (0.597)	1.488 (0.321)	10.089 (0.888)	8.478 (1.696)	4.18 (0.817)
Kurtosis	2.897 (0.519)	2.902 (0.527)	2.689 (0.694)	2.898 (0.558)	2.890 (0.523)	2.892 (0.772)
MSE	- (-)	2.405 (1.733)	5.158 (0.977)	- (-)	104.463 (28.519)	86.095 (16.258)
MAE	- (-)	1.185 (0.386)	1.809 (0.178)	- (-)	8.136 (1.101)	7.403 (0.735)

5 Empirical application

In this Section, we show how temporal disaggregation can be used in a real-world problem.

The urgent need to address climate change has propelled the scientific community to explore innovative approaches to quantify and manage greenhouse gas (GHG) emissions. Carbon intensity, a crucial metric that measures the amount of carbon dioxide equivalent emitted per unit of economic activity (e.g. sales), plays a pivotal role in assessing the environmental sustainability of industries, countries, and global economies. By focusing on emissions per unit of economic output, carbon intensity accounts for the fact that larger organizations or economies may naturally produce more emissions simply due to scale. This allows for a fair comparison of sustainability performance across different entities, regardless of size. Accurate and timely carbon accounting and the development of robust measurement frameworks are essential for effective emission reduction strategies and the pursuit of sustainable development goals. While carbon accounting frameworks offer valuable insights into emissions quantification, they are not without limitations. One of those limitations is the frequency with which this information is released, generally at an annual frequency, while most companies' economic indicators are made public on a quarterly basis. This is a perfect example in which temporal disaggregation can be used to bridge the gap between data availability and prompt economic and financial analyses. In this application, the variable of interest is the GHG emissions for IBM between Q3 2005 and Q3 2021, at annual frequency, resulting in 17 datapoints, i.e. $Y \in \mathbb{R}^{17}$. For the high-frequency data, we used the balance sheet, income statement, and cash flow statement quarterly data between Q3 2005 and Q3 2021, resulting in 68 datapoints for the 128 variables. We remove variables that have a pairwise correlation higher than 0.99, resulting in a filtered dataset with 112 variables, i.e. $X \in \mathbb{R}^{68\times 112}$. In this example, we employed the adaptive LASSO method (`method = "adaptive-spTD"`) as a way to select the best variables that can be used to recover the high-frequency observations and we applied the `aggMat = "sum"` aggregation method. The rationale for using this method, in conjunction with `aggRatio = 4` is to ensure that the disaggregated quarterly carbon intensity values are consistent with the overall annual figures as the goal is to break down the emissions and sales data such that the sum of the quarterly carbon intensities equals the yearly total. The adaptive LASSO procedure select only seven non-zero coefficients, which are Actual sales per employee, trailing 12-month net sales, Enterprise value, Current liabilities, total liabilities and equity, total line of credit and net debt. Actual sales per employee, trailing 12-month net sales and Enterprise value indicate company sales and size, both linked to the company's economic activity, operational intensity, and commitment to sustainability, and hence potential emissions. Current liabilities and total liabilities and equity reflect the company's financial position and operational scale, both of which might influence emissions. Total

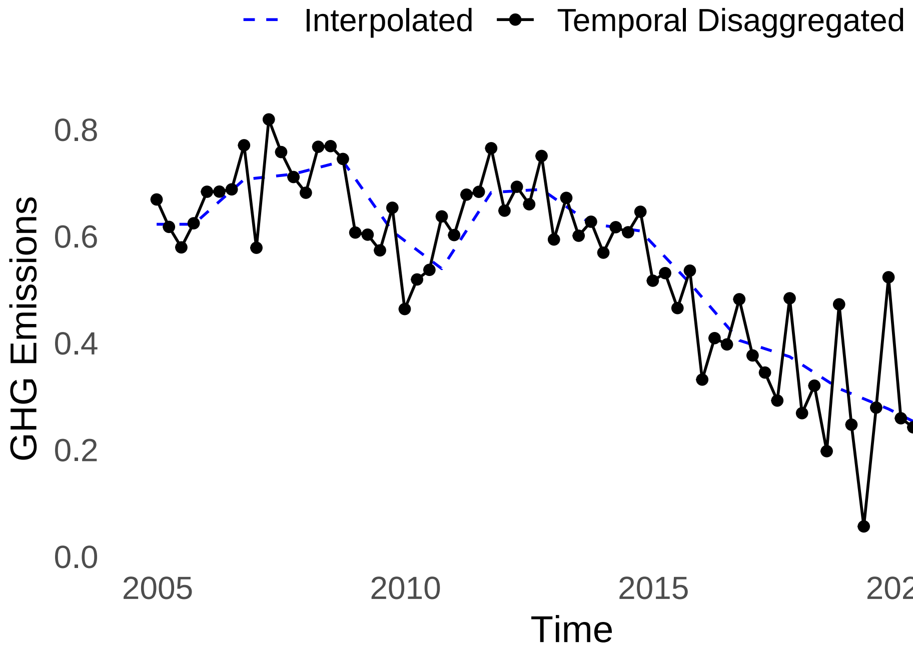

Figure 5: Temporal disaggregated and interpolated GHG emissions observations. In this example, we used the setting `aggMat = sum`, so the sum of the quarterly disaggregated GHG sums to the actual annual observation.

line of credit and net debt highlight the company's ability to borrow, which could impact investment in emissions-reducing projects. We show the results in Figure 5 alongside a linear interpolation method.

As it is possible to observe from the plot, the interpolated data do not fluctuate as we would expect from real GHG emissions, as the method is not conditional on the variability of the high-frequency variables. In this respect, the temporal disaggregated observations show more realistic dynamics. This result can then be used to compute the GHG intensity, computing the ratio between GHG emissions and the sales for the corresponding quarter.

6 Summary

In this paper, we have given an overview of the key functionality for the **DisaggregateTS** R package. The package builds on features of the existing **tempdisagg** package allowing the user to easily apply the regularized Chow-Lin type procedure of Mosley et al. (2022) and compare this with classical methods based on interpolation via smoothing (e.g. Denton, 1971), or the Fernández (1981) and Fernández (1981) methods in the case where series are not co-integrated. The package may be extended in future to allow cross-sectional constraints, the disaggregation of panel time-series data, and the accommodation of factor (latent-variable) structures in the temporal-disaggregation problem.

It is important to point out that when performing disaggregation, one should be careful to select an appropriate set of indicator time-series, and not purely rely on the model-selection procedures deployed here. Whilst these methods can help pick from a set of indicators, there is still room for them to pick irrelevant/spurious series, especially given the short nature of the aggregate series. To this end, we do not recommend just relying on one method alone, but rather a user deploy several methods of disaggregation. The user should then attempt to obtain some form of (potentially qualitative) external validation of the resultant series in their application of choice. One should always remember, that as the high-frequency series is not observed, there can be no direct empirical validation of the methods other than at the aggregate scale.

References

A. Belloni and V. Chernozhukov. Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2):521–547, 2013. [p65]

J. Bournay and G. Laroque. Réflexions sur la méthode d'elaboration des comptes trimestriels. In *Annales de l'INSEE*, volume 36, pages 3–30. JSTOR, 1979. [p64, 65]

P.-A. Cholette. Adjusting sub-annual series to yearly benchmarks. *Survey Methodology*, 10(1):35–49, 1984. [p65]

G. C. Chow and A.-l. Lin. Best linear unbiased interpolation, distribution, and extrapolation of time series by related series. *The review of Economics and Statistics*, 53(4):372–375, 1971. [p63, 64, 65, 66, 67]

G. C. Chow and A.-L. Lin. Best linear unbiased estimation of missing observations in an economic time series. *Journal of the American Statistical Association*, 71(355):719–721, 1976. [p63]

A. Cottrell and R. Lucchetti. *Gretl User's Guide: Gnu Regression, Econometrics and Time-series Library*, 2023. URL https://gretl.sourceforge.net/gretl-help/gretl-guide.pdf. Accessed: 2023-10-08. [p65]

E. B. Dagum and P. A. Cholette. *Benchmarking, Temporal Distribution, and Reconciliation Methods for Time Series*. Springer, 2006. [p62, 63, 66]

F. T. Denton. Adjustment of monthly or quarterly series to annual totals: an approach based on quadratic minimization. *Journal of the american statistical association*, 66(333):99–102, 1971. [p63, 65, 71]

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2): 407–451, 2004. [p65]

R. B. Fernández. A methodological note on the estimation of time series. *The Review of Economics and Statistics*, 63(3):471–476, 1981. [p63, 65, 71]

P. Fuleky. *Macroeconomic forecasting in the era of big data: Theory and practice*, volume 52. Springer, 2019. [p62]

R. B. Litterman. A random walk, markov model for the distribution of time series. *Journal of Business & Economic Statistics*, 1(2):169–173, 1983. [p63, 65]

L. Mosley and K. S. Nobari. *DisaggregateTS: High-Dimensional Temporal Disaggregation*, 2024. URL https://CRAN.R-project.org/package=DisaggregateTS. R package version 3.0.1. [p62]

L. Mosley, I. A. Eckley, and A. Gibberd. Sparse Temporal Disaggregation. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 185(4):2203–2233, 10 2022. ISSN 0964-1998. doi: 10.1111/rssa.12952. URL https://doi.org/10.1111/rssa.12952. [p62, 63, 65, 66, 71]

E. M. Quilis. Temporal disaggregation of economic time series: The view from the trenches. *Statistica Neerlandica*, 72(4):447–470, 2018. [p64]

C. Sax and P. Steiner. Temporal disaggregation of time series. *The R Journal*, 5(2):80–87, 2013. [p62, 65, 66]

C. Sax, P. Steiner, and T. Di Fonzo. *'tempdisagg': Methods for Temporal Disaggregation and Interpolation of Time Series*, 2023. URL https://CRAN.R-project.org/package=tempdisagg. R package version 1.1.1. [p62, 64, 66]

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978. [p65]

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. [p63]

S. Van de Geer, P. Bühlmann, and S. Zhou. The adaptive and the thresholded lasso for potentially misspecified models (and a lower bound for the lasso). *Electronic Journal of Statistics*, 5:688–749, 2011. [p67]

M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019. [p62]

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101 (476):1418–1429, 2006. [p66]

Luke Mosley
Lancaster University
School of Mathematical Sciences
Lancaster, United Kingdom
lukemos313@gmail.com

Kaveh Salehzadeh Nobari
Imperial College LondonLondon School of Economics and Political Science
Centre for Climate Finance & Investment, London, United Kingdom
Department of Psychological and Behavioural Science, London, United Kingdom
k.salehzadeh-nobari@imperial.ac.uk

Giuseppe Brandi
Imperial College LondonNortheastern University London
Centre for Climate Finance & Investment, London, United Kingdom
Khoury College of Computer Sciences, London, United Kingdom
g.brandi@imperial.ac.uk

Alex Gibberd
Lancaster University
School of Mathematical Sciences
Lancaster, United Kingdom
a.gibberd@lancaster.ac.uk