Trend filtering for functional data

Tomoya Wakayama1 | Shonosuke Sugasawa2

1Graduate School of Economics, The University of Tokyo, Bunkyo, Tokyo, 113-8654, Japan
2Faculty of Economics, Keio University, Minato, Tokyo, 108-8345, Japan

Correspondence
Tomoya Wakayama, The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo 113-8654, Japan.
Email: tom-w9@ecc.u-tokyo.ac.jp

Funding information
Japan Society for the Promotion of Science, Grant/Award Numbers: 18H03628, 20H00080, 21H00699

Despite increasing accessibility to function data, effective methods for flexibly estimating underlying functional trend are still scarce. We thereby develop a functional version of trend filtering for estimating trend of functional data indexed by time or on general graph by extending the conventional trend filtering, a powerful nonparametric trend estimation technique for scalar data. We formulate the new trend filtering by introducing penalty terms based on $L_2$-norm of the differences of adjacent trend functions. We develop an efficient iteration algorithm for optimizing the objective function obtained by orthonormal basis expansion. Furthermore, we introduce additional penalty terms to eliminate redundant basis functions, which leads to automatic adaptation of the number of basis functions. The tuning parameter in the proposed method is selected via cross validation. We demonstrate the proposed method is locally adaptive and can identify change points through simulation studies and applications to real-world datasets.

KEYWORDS
ADMM algorithm, functional time series data, group fused lasso, spatial functional data, trend estimation on graphs

1 INTRODUCTION

Due to advances in measurement devices and data storage resources, it is nowadays possible to observe functions as realizations of random experiments, and thus functional data analysis (FDA) has expanded rapidly in recent decades. Functional versions for many branches of statistics have been provided, for example, in Ramsay (2004), Kokoszka and Reimherr (2017), and Horváth and Kokoszka (2012).

The conventional techniques of FDA for independent functional data have been recently extended to dependent situations (both time series and spatial cases). In fact, for functional time series data, standard stationary models for multivariate data have been extended (e.g., Besse et al., 2000; Gao et al., 2019; Hörmann et al., 2013, 2015; Klepisch et al., 2017; Klepisch & Klüppelberg, 2017) and theoretical properties have also been widely investigated (e.g., Aue et al., 2017; Aue & Klepisch, 2017; Bosq, 2000; Cerovecki et al., 2019; Kühnert, 2020; Spangenberg, 2013). On the other hand, effective estimations of functional trend under nonstationary situations are not well developed despite their importance in real applications. van Delft and Eichler (2018) addressed a framework for locally stationary functional times series, but its flexibility for trend estimation is still limited. Regarding spatial functional data, while spatial interpolation methods under spatial stationarity have been developed (e.g., Giraldo et al., 2011; Nerini et al., 2010), there are some attempts to estimate nonstationary spatial trend determined by some external covariates (e.g., Caballero et al., 2013; Menafoglio et al., 2013, 2016). However, many flexible estimation methods for spatially varying trend are not considered.

Although many useful tools are available in FDA, the flexibility of existing methods may be limited; that is, handling abrupt changes in a trend is challenging. Hence, the need for locally adaptive smoothing methods arises. For univariate time series, locally adaptive methods flourish, and their theoretical properties have been widely discussed (Mammen & van de Geer, 1997; Steidl et al., 2006). Among them, trend filtering

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2023 The Authors. Stat published by John Wiley & Sons Ltd.
a nonparametric method, is prominent because of its flexibility and local adaptability. Also its theoretical justifications (e.g., minimax convergence rate) have been provided (Tibshirani, 2014; Wang et al., 2014). Additionally, Wang et al. (2016) extended trend filtering to spatial data, which enables us to estimate spatial trend with abrupt changes revealed.

In this work, we provide an effective local smoothing method for functional time series data by extending trend filtering for scalar data. To solve the optimization problem for functional trend filtering, we expand the functional data via orthonormal basis functions and transform the objective function. We find that this transformed objective function is a mixture of the fused lasso (Tibshirani et al., 2005) and the grouped lasso (Lounici et al., 2011; Tibshirani, 1996; Yuan & Lin, 2006). This is rather different from the case of scalar, where only fused lasso-like penalties are considered, so that existing algorithms for regularized methods related to lasso cannot be directly applied. We then develop an iterative algorithm based on the idea of ADMM (Boyd et al., 2011; Ramdas & Tibshirani, 2016), in which each updating procedure can be easily carried out, and solve the difficulty. To specify an initial value in the ADMM algorithm, we develop a functional version of HP filtering. We also compare the HP filtering with the proposed functional trend filtering to clarify the importance of using the sparsity-inducing penalization approach. Furthermore, for automatic selection of the optimal number of basis functions, we introduce additional penalty terms to omit redundant basis functions, which will be noted by sparse functional trend filtering. The resulting objective function can also be optimized by a slight modification of the ADMM algorithm. We also extend functional trend filtering for time series data (i.e., data on linear chain graph) to data on a general graph, which enables us to apply the proposed smoothing method to spatial functional data. As for the tuning parameter selection, we suggest using cross validation, which is fairly feasible due to the efficient optimization algorithm.

The remainder of the paper is organized as follows. Section 2 offers a brief review of trend filtering and its periphery, which is deeply related to our work. In Section 3, we present the methods, functional trend filtering, for both functional time series and spatial data, and describe the algorithm to carry out the proposed method. Also, we discuss the selection of the number of basis functions. In Section 4, we compare the proposed method with some existing approaches through simulation studies. In Section 5, we apply the proposed method to functional time series (fertility rates as a function of age in each year) and functional spatial data (the number of confirmed COVID-19 cases as a function of day in Japanese prefectures). Finally, we conclude with a discussion in Section 6.

### 2 | Review of Trend Filtering for Scalar Data

Before describing the proposed methods for functional data, we briefly review trend filtering, a powerful tool for locally adaptive smoothing for scalar time series data. Let $y_1, ..., y_T$ be a sequence of observations, and we are interested in denoising the observations to estimate the underlying trend denoted by $\beta = (\beta_1, ..., \beta_T)^T$. The $k$th order trend filtering (Kim et al., 2009; Tibshirani, 2014) is defined as the minimizer of the following objective function:

$$
\frac{1}{2} \sum_{t=1}^{T} (y_t - \beta_t)^2 + \lambda \sum_{t=1}^{T-k+1} |\Delta_t \beta_t|,
$$

(1)

where $\lambda \geq 0$ is a tuning parameter that controls the trade-off between the fit to the observed data and smoothing the trend estimation. Here, $\Delta_t$ is the $t$th row vector of the $k$th order discrete difference operator matrix $\Delta^{(k)}$, defined as

$$
\Delta^{(k)} := \begin{cases} 
D^{(0)} & \text{for } k = 0, \\
D^{(k)} \Delta^{(k-1)} & \text{for } k \geq 1,
\end{cases}
$$

where $D^{(k)}$ is the following $(T-k+1) \times (T-k)$ matrix:

$$
D^{(k)} := \begin{pmatrix}
1 & -1 & 0 & \ldots & 0 & 0 \\
0 & 1 & -1 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 1 & -1
\end{pmatrix}.
$$

For example, (1) with $k = 0$,

$$
\frac{1}{2} \sum_{t=1}^{T} (y_t - \beta_t)^2 + \lambda \sum_{t=1}^{T-1} |\beta_t - \beta_{t+1}|,
$$

(2)
is the same form of fused lasso (Tibshirani et al., 2005) and the penalty makes many differences to zero exactly and leave others nonzero values, leading to piece-wise constant estimation of $\beta$. In general, sparsity of $\beta$ under kth order discrete difference operator matrix suggests that the estimated components have a specific kth order piece-wise polynomial structure (Tibshirani, 2014). While trend filtering is a locally adaptive estimator defined by a regularization problem with a nonsmooth penalty, it is still computationally efficient owing to its convexity.

Trend filtering is also applicable to spatial data (Wang et al., 2016). Let $V = \{1, ..., n\}$ be a set of sample index, which can be regarded as the vertex of graph $G = (V,E)$. Here, $E = \{e_1, ..., e_m\}$ is a set of undirected edges according to the spatial adjacent structure, where $e_h \in V \times V$ for $h = 1, ..., m$ and $m$ is the total number of adjacency relationships. For example, $e_h = (i,j)$ means that $ih$ and $j$th locations are adjacent. Let $\Delta^{(0)} \in \{1,0,-1\}^{m \times n}$ be the oriented incidence matrix of the graph $G$, that is, $\Delta_N^{(0)} = -1$, $\Delta_N^{(0)} = -1$ and the other elements in the $ih$ row vector of $\Delta^{(0)}$ is $0$ if $e_h = (i,j)$. We then define

$$\Delta^{(k+1)} := \begin{cases} (\Delta^{(0)})^T \Delta^{(k)} & \text{for even } k, \\ \Delta^{(0)} \Delta^{(k)} & \text{for odd } k. \end{cases}$$

This $\Delta^{(k)}$ is hereinafter referred to as the $k$th order graph difference operator matrix. The $k$th order spatial trend filtering estimate is obtained as the minimizer of the following function:

$$\frac{1}{2} \sum_{i=1}^{n} (y_i - \hat{\beta}_i)^2 + \lambda \|\Delta^{(k)} \beta\|_1,$$

where $\lambda$ is a tuning parameter. We remark it is also a form of fused lasso and accordingly it can be solved by basic convex optimization algorithms. Wang et al. (2016) discusses the computational aspect in detail. Notably, the penalty term encourages sparsity in graph differences in trend, which yields a piece-wise polynomial nature of the estimator as the original trend filtering (1).

## 3 FUNCTIONAL TREND FILTERING

We will develop the method discussed above into functional data, that is, find a trend among functions.

### 3.1 Settings and objective function

Let $(\Omega, \mathcal{A}, P)$ be an arbitrary probability space. The space $L^2(\mathcal{X})$ is defined as the set of all real-valued square integrable functions on a compact set $\mathcal{X} \subset \mathbb{R}$. It is a Hilbert space with norm $\|f\|_2 = \left(\int_{\mathcal{X}} f^2(x) dx\right)^{1/2}$, induced by the inner product $(f,g) = \int_{\mathcal{X}} f(x)g(x) dx$ for $f, g \in L^2(\mathcal{X})$. We consider a serially indexed collection $\{Y_t(\cdot) : t = 1, ..., T\}$ of random functions defined on the same probability space: $Y_t : \Omega \rightarrow L^2(\mathcal{X})$ is a measurable map. $\{Y_t(\cdot) : t = 1, ..., T\}$ denotes a set of observed functions. The trend of $\beta_t = E[Y_t]$ as a function of $t$ is of interest. To estimate $\beta = (\beta_1, ..., \beta_T) \in (L^2(\mathcal{X}))^T$, we propose the $k$th order functional trend filtering defined as

$$\hat{\beta}_{TF}^{(k)} = \arg \min_{\beta} \frac{1}{2} \sum_{t=1}^{T} \left( \int_{\mathcal{X}} (y_t(x) - \beta_t(x))^2 dx + \lambda \sum_{i=1}^{T-k} \left( \int_{\mathcal{X}} \Delta_t^{(k)} \beta(x) \right)^2 dx \right)^{1/2},$$

where $\lambda > 0$ is a tuning parameter and $\Delta_t^{(k)}$ is the $t$th row vector of the $k$th order difference operator matrix $\Delta^{(k)}$. For instance, if $k = 0$, the penalty is $\sum_{t=1}^{T} \|\beta_t - \beta_{t+1}\|_2$, which can be regarded as the functional version of the group fused lasso (Alaiz et al., 2013). Henceforth, we have to solve the functional version of the group fused lasso with general order $k$, desiring that some elements of $\{\Delta_t^{(k)} \beta_{TF}^{(k)} : t = 1, ..., T-k\}$ are set to zero.

Because the observations and the true functions are infinite dimensions and difficult to handle, we first prepare $L$ orthonormal basis functions $\{\varphi_{\ell} : \ell = 1, ..., L\}$ on $L^2(\mathcal{X})$, which satisfy

$$\langle \varphi_{\ell}, \varphi_{\ell'} \rangle = \begin{cases} 1 & \text{if } \ell = \ell', \\ 0 & \text{otherwise}. \end{cases}$$

Then, using the approximate expansions $y_t(x) \approx \sum_{\ell=1}^{L} z_{t\ell} \varphi_{\ell}(x)$ and $\beta_t(x) \approx \sum_{\ell=1}^{L} b_{t\ell} \varphi_{\ell}(x)$ with $z_{t\ell} = \langle y_t, \varphi_{\ell} \rangle$ and $b_{t\ell} = \langle \beta_t, \varphi_{\ell} \rangle$ for all $t$ and $\ell$, we reduce problem (3) to minimization of the objective function.
with respect to \( \mathbf{b}_r = (b_{1r},...,b_{Tr})^\top \in \mathbb{R}^T \). Define \( B = (b_1,...,b_L) \) and \( z_r = (z_{1r},...,z_{Tr})^\top \). The above objective function can be further rewritten as

\[
\frac{1}{2} \sum_{r=1}^L \sum_{t=1}^T (z_{tr} - b_{tr})^2 + \lambda \sum_{t=1}^{T-1} \left[ \sum_{r=1}^L (\Delta^{(t)} b_r)^2 \right]^{1/2},
\]

where \( \| \cdot \|_2 \) denotes the \( \ell^2 \) norm. Focusing on the latter half, we notice it is a mixture of \( k \)th order fused lasso type penalty and grouped lasso type penalty. In the case of scalar (Kim et al., 2009; Tibshirani, 2014), it is enough to consider fused lasso-like penalty, but in the case of functional data, mixture of group lasso and fused lasso-like penalty is necessary. It is natural to use group lasso for the \( k \)th order differences of coefficient vectors because the functional trend filtering aims to shrink the \( k \)th order functional difference toward the function identically equal to 0.

To clarify this issue, assume \( k = 0 \) here. Then (4) is represented as

\[
\frac{1}{2} \sum_{r=1}^L \sum_{t=1}^T (z_{tr} - b_{tr})^2 + \lambda \sum_{t=1}^{T-1} \| b_{t+1} - b_t \|_2,
\]

where \( \mathbf{b}_t = (b_{1t},...,b_{Lt})^\top \in \mathbb{R}^L \). This problem is different from the scalar version (2) in that problem (2) is simply rewritten as

\[
\arg\min_{\beta \in \mathbb{R}^L, \alpha \in \mathbb{R}^T} \frac{1}{2} \| y - \beta \|_2^2 + \lambda \| \alpha \|_1 \quad \text{subject to } \alpha = \Delta^{(k)} \beta,
\]

where \( y = (y_1,...,y_T)^\top \), and it is the very form in which ADMM is employed. Because the first \( L \) terms and last \( T - 1 \) terms in (5) do not have the same vector in common, it is challenging to apply vanilla ADMM (Boyd et al., 2011; Ramdas & Tibshirani, 2016). Accordingly, solving this problem is not a straightforward extension of the scalar version.

### 3.2 | Optimization

For notational simplicity, we use \( \Delta \) instead of \( \Delta^{(k)} \) in what follows. To optimize (4), we first introduce two unit vectors, \( \mathbf{e}_t^T \in \mathbb{R}^{T-k-1} \) and \( \mathbf{e}_t^T \in \mathbb{R}^k \), whose only \( t \)th and \( r \)th elements are 1, respectively, and the other elements are 0.

There is no simple relationship between \( \mathbf{b}_r \) and \( \Delta \mathbf{b}_r \) that appears in the first \( L \) terms and the last \( T - k - 1 \) terms of Equation (4), but we have \( L \times (T - k - 1) \) relationships: \( \Delta \mathbf{b}_t^k = (\mathbf{e}_t^T)^\top \Delta \mathbf{b}_r \) for \( r = 1,...,L, t = 1,...,T - k - 1 \). Hence, by incorporating them, we rewrite the optimization of (4) as the following constraint optimization problem:

\[
\arg\min_{\mathbf{a}, \mathbf{b} \in \mathbb{R}^L, \mathbf{e} \in \mathbb{R}^T} \frac{1}{2} \sum_{r=1}^L \sum_{t=1}^T (z_{tr} - b_{tr})^2 + \lambda \sum_{t=1}^{T-1} \| \mathbf{a}_t \|_2
\]

subject to \( \mathbf{a} \mathbf{e}_t^k = (\mathbf{e}_t^T)^\top \Delta \mathbf{b}_r \), for \( r = 1,...,L, t = 1,...,T - k - 1 \).

Note that the constraint is not a simple form but the objective function is similar to one for alternating direction method of multipliers (ADMM) algorithm (Boyd et al., 2011; Ramdas & Tibshirani, 2016), which breaks the problem into smaller pieces that are easier to deal with. We then define an augmented Lagrangian function

\[
L_r((\mathbf{a}_r), (\mathbf{b}_r), (u_r)) := \frac{1}{2} \sum_{r=1}^L \sum_{t=1}^T (z_{tr} - b_{tr})^2 + \lambda \sum_{t=1}^{T-1} \| \mathbf{a}_t \|_2 + \sum_{r=1}^L \sum_{t=1}^{T-1} u_{tr} \left( (\mathbf{e}_t^T)^\top \Delta \mathbf{b}_r - \mathbf{a}_t \mathbf{e}_t^k \right) + \rho \sum_{r=1}^L \sum_{t=1}^{T-1} \left( (\mathbf{e}_t^T)^\top \Delta \mathbf{b}_r - \mathbf{a}_t \mathbf{e}_t^k \right)^2,
\]

where \( u_{tr} \) is Lagrange multipliers and \( \rho > 0 \) controls the influence of the violation of equality constraint.

Because there is no apparent closed form solution for \( \mathbf{b}_r \), minimizing the objective function \( L_r \), we develop an algorithm consisting of the iterations:
\[ \mathbf{b}_{\ell}^{(v+1)} := \arg \min_{\mathbf{b}_{\ell}} \left( \{ \mathbf{a}_t^{(v)} \}, \{ \mathbf{b}_{\ell}^{(v+1)} \}, \{ \mathbf{u}_{\ell t}^{(v)} \} \right) \]

\[ \mathbf{a}_t^{(v+1)} := \arg \min_{\mathbf{a}_t} \left( \{ \mathbf{a}_t^{(v)} \}, \{ \mathbf{b}_{\ell}^{(v+1)} \}, \{ \mathbf{u}_{\ell t}^{(v)} \} \right) \]

\[ \mathbf{u}_{\ell t}^{(v+1)} := \mathbf{u}_{\ell t}^{(v)} + \rho \left( \mathbf{e}_t^{(v+1)} \right) - \mathbf{b}_{\ell}^{(v+1)} \mathbf{e}_t^{(v+1)}. \]

Remark that the updating step can be carried out in parallel for each \( \ell = 1, \ldots, L \) as dual decomposition method. This is the case for the updating steps of \( \mathbf{a}_t \) and \( \mathbf{u}_{\ell t} \). We implement the iterations until the values converge. The steps of \( \mathbf{b}_{\ell} \) and \( \mathbf{a}_t \) are specifically described as follows:

- **(update \( \mathbf{b}_{\ell} \))** For given \( \mathbf{u}_{\ell t} \) and \( \mathbf{a}_t \), we update \( \mathbf{b}_{\ell} \) as the minimizer of \( L_{\rho} \), which is a quadratic function of \( \mathbf{b}_{\ell} \). Because its derivative with respect to \( \mathbf{b}_{\ell} \) is given by

  \[ (I + \rho \Delta^T \Delta) \mathbf{b}_{\ell} - \mathbf{z}_\ell + \sum_\ell \mathbf{u}_{\ell t} \left( \mathbf{e}_t^{(v)} \right)^T \Delta - \rho \sum_\ell \left( \mathbf{e}_t^{(v)} \right)^T \mathbf{a}_t \mathbf{e}_t^{(v)}, \]

  the minimizer can be obtained as

  \[ (I + \rho \Delta^T \Delta)^{-1} \left\{ \mathbf{z}_\ell - \sum_\ell \mathbf{u}_{\ell t} \left( \mathbf{e}_t^{(v)} \right)^T \Delta + \rho \sum_\ell \left( \mathbf{e}_t^{(v)} \right)^T \mathbf{a}_t \mathbf{e}_t^{(v)} \right\}. \]

- **(update \( \mathbf{a}_t \))** For given \( \mathbf{u}_{\ell t} \) and \( \mathbf{b}_{\ell} \), \( \mathbf{a}_t \) is updated as the minimizer of

  \[ \left( I \sum_\ell ||\mathbf{a}_t||_2 + \rho \sum_\ell \sum_\ell \left( \mathbf{e}_t^{(v)} \right)^T \Delta \mathbf{b}_{\ell} - \mathbf{a}_t \mathbf{e}_t^{(v)} + \frac{\mathbf{u}_{\ell t}}{\rho} \right)^2. \]

Because this function is nondifferentiable, we employ a proximal method. We denote the first half (nonsmooth part) by \( f_{\text{non}} \) and the second half (smooth part) by \( f_{\text{sm}} \). Because \( f_{\text{non}} \) is convex and \( \nabla \mathbf{a} f_{\text{sm}} \) is Lipschitz continuous, we can apply FISTA (Beck & Teboulle, 2009). The proximity operator of \( \lambda \| \cdot \|_2 \) is known as \( S_\lambda \left( s \right) = \max \left( 0, 1 - \lambda \| s \|_2 \right) s \). Then, the update procedure is as follows.

**Algorithm 1** Update \( \mathbf{a}_t \) by FISTA.

1. Initialize \( \mathbf{w}_0, s_0 = 1 \) and \( j = 0 \).
2. \( \mathbf{w}_{j+1} = S_{j} \left( \mathbf{w}_j - \nabla \mathbf{a} f_{\text{sm}} \left( \mathbf{w}_j \right) \right) \)
3. \( s_{j+1} = \left( 1 + \sqrt{1 + 4 s_j^2} \right) / 2 \)
4. \( \mathbf{w}_{j+1} = \mathbf{w}_{j+1} + (s_j - 1) \left( \mathbf{w}_{j+1} - \mathbf{w}_j \right) / s_{j+1} \)
5. If the algorithm has converged, we set \( \mathbf{a}_t^{(v+1)} = \mathbf{w}_j \). Otherwise, set \( j = j + 1 \) and go back to Step 2.

Thus, the whole procedure of the iterative algorithm is summarized as follows.

**Algorithm 2** Functional trend filtering.

1. Initialize \( v = 0, \{ \mathbf{a}_t^{(v)} \}, \{ \mathbf{b}_{\ell}^{(v)} \}, \{ \mathbf{u}_{\ell t}^{(v)} \} \) and \( \rho \).
2. Update \( \mathbf{b}_{\ell}^{(v+1)} \) as (8) for all \( \ell \).
3. Update \( \mathbf{a}_t^{(v+1)} \) via Algorithm 1 for all \( t \).
4. Update \( \mathbf{u}_{\ell t}^{(v+1)} \) as (7) for all \( \ell \) and \( t \).
5. If the algorithm has converged, the algorithm is terminated. Otherwise, set \( j = j + 1 \) and go back to Step 2.

The convergence of Algorithm 2 is empirically confirmed. We suggest choosing \( \rho \) to be proportional to \( \lambda \) in order to stabilize the result (Ramdas & Tibshirani, 2016).
Using the coefficients $b^\text{TF}_t$ computed by the procedure, we obtain the function

$$
\hat{\beta}^\text{TF}_t(x) = \sum_{r=1}^{L} b^\text{TF}_t \phi_r(x),
$$

for $t = 1, \ldots, T$, which is the estimator of the trend.

Next, we construct a simplified version of functional trend filtering as an alternative smoother:

$$
\hat{\beta}^\text{HP} = \arg\min_{\beta} \frac{1}{2} \sum_{t=1}^{T} \int \left( y_t(x) - \hat{\beta}(x) \right)^2 dx + \lambda \sum_{t=1}^{T} \int \left( \Delta_t \hat{\beta}(x) \right)^2 dx.
$$

The difference between (9) and (3) is the penalty. Specifically, the penalty in (9) is the squared value of the $L^2$-norm used in (3). Because the estimator defined in (9) can be regarded as an extension of Hodrick-Prescott (HP) filter (Hodrick & Prescott, 1997), we refer to this method as functional HP filter. Although the squared norm penalty does not produce sparsity in the differences, the method is easy to implement. In fact, by expanding (9) via orthonormal functions, we have the following approximation of the objective function:

$$
\frac{1}{2} \sum_{r=1}^{L} \left\| z_r - b^r \right\|^2_2 + \lambda \sum_{t=1}^{T} \sum_{r=1}^{L} (\Delta_t b^r)^2,
$$

which yields the closed form solution given by

$$
\hat{b}^\text{HP}_r = \left( I_T + 2 \sum_{t=1}^{T-1} \lambda \Delta_t \Delta_t^T \right)^{-1} z_r,
$$

$r = 1, \ldots, L$. Because the implementation of the method is fast due to the presence of an analytical solution, we can use $\hat{b}^\text{HP}_r$ as the initial values $b^{(0)}_r$ in Algorithm 2.

Finally, we discuss the choice of tuning parameter $\lambda$. We suggest selecting the optimal value of $\lambda$ via “deterministic” $K$-fold cross validation (CV), described as follows.

**Algorithm 3 Deterministic $K$-fold CV.**

1. Initialize $i = 1$ and $E = 0$
2. Extract every $K$th function. That is, extract $\{ y_{i+nk} : n \in \mathbb{N}, i+nk \leq T \}$ from $\{ y_t : t = 1, \ldots, T \}$ and define the set as test data. Set the remaining as training data and implement filtering.
3. Interpolate the test data by estimates; namely, $y_t$ in test data is interpolated by the midpoint of $\hat{y}_{t-1}$ and $\hat{y}_{t+1}$.
4. $E = E + \sum_{t=\text{test}} \int_X \left( y_t(x) - \hat{y}_t(x) \right)^2 dx$
5. If $i < K$, $i = i + 1$ and go back to step 2. Otherwise, $K$-fold CV is terminated and output is the value of $E$.

For a pre-specified set of candidate values of $\lambda$, we compute the interpolation error via the above CV method and select the optimal value as the minimizer of the interpolation error. Although we will not go into the detail, the optimal value of the order of difference, $k$, can also be selected by the $K$-fold CV.

### 3.3 Extension to functional data on graph

Let $G = (V, E)$ be the graph with vertices $V = \{1, \ldots, n\}$ and undirected edges $E = \{e_1, \ldots, e_m\}$, representing spatial adjacent structure. Let $\{ Y_{i}(:) : 1, \ldots, n \}$ be random functions on the vertices, which take values in the space $L^2(\mathcal{X})$ on a compact set $\mathcal{X} \subset \mathbb{R}$. Suppose that $E[Y_i(x)] = \beta_i(x)$ for $i = 1, \ldots, n$ and we are interested in the estimation of $\hat{\beta}_i(x)$. Let $\Delta^{(k)}$ be $k$th order graph difference operator matrix defined in Section 2. We propose the $k$th order functional trend filtering on graph to estimate $\beta = (\beta_1, \ldots, \beta_n)^\top$ by
\[
\hat{\beta}^{\text{TF}} = \arg \min_{\beta} \frac{1}{2} \sum_{i=1}^{n} \int (y_i(x) - \hat{\beta}(x))^2 \, dx + \lambda \sum_{p=1}^{q} \left[ \int (\Delta_{p}^{(k)} \beta(x))^2 \, dx \right]^{1/2}
\]

where \( q = n \) if \( k \) is odd, \( q = m \) otherwise. The penalty quantifies how much \( \beta \) vary locally in the sense of \( k \)th order graph differences. We prepare \( L \) orthonormal basis functions \( \phi_1(x), \phi_2(x), \ldots, \phi_q(x) \) and approximate \( y_i(x) \approx \sum_{\ell=1}^{L} z_{\ell} \phi_{\ell}(x) \) and \( \beta_i(x) \approx \sum_{\ell=1}^{L} b_{\ell} \phi_{\ell}(x) \) with \( z_{\ell} = \langle y_i, \phi_{\ell} \rangle \) and \( b_{\ell} = \langle \beta_i, \phi_{\ell} \rangle \) for all \( i \) and \( \ell \). This is an extension of Wang et al. (2016) to functional data, but the optimization is far more complicated, as we showed in time series.

In the following discussion in this section, we write \( \Delta \) for \( \Delta_{p}^{(k)} \). Define two standard unit vectors \( e^0_\ell \in \mathbb{R}^d \) and \( e^1_\ell \in \mathbb{R}^d \), whose only \( t \)th and \( t' \)th elements are 1, respectively, and the other elements are 0, and \( b_{\ell} = (b_{\ell}, \ldots, b_{\ell})^T \in \mathbb{R}^d \). Following the same logic as the previous section, we regard the problem to find (11) as a problem to get \( \{b_{\ell}\} \) minimizing an augmented Lagrangian, for a parameter \( \rho > 0 \).

\[
L_{\rho}(\{a_{\ell}\}, \{b_{\ell}\}, \{u_{\ell}\}) := \frac{1}{2} \sum_{\ell=1}^{L} \|z_{\ell} - b_{\ell}\|^2 + \lambda \sum_{p=1}^{q} \|a_{p}\|^2 + \sum_{p=1}^{q} \sum_{\ell=1}^{L} u_{\ell} \psi((e^0_\ell)^T \Delta b_{\ell} - \psi_{p} e^0_\ell)^2 + \frac{\rho}{2} \sum_{p=1}^{q} \sum_{\ell=1}^{L} \|e^0_\ell\|^2 \Delta b_{\ell} - \psi_{p} e^0_\ell)^2.
\]

To solve this problem, we can again utilize Algorithm 1. Using the acquired coefficients \( b_{\ell}^{TF} \) computed by the procedure, we obtain the function

\[
\hat{\beta}^{\text{TF}}(x) = \sum_{\ell=1}^{L} b_{\ell}^{TF} \phi_{\ell}(x),
\]

for \( i = 1, \ldots, n \). This is the estimator of the proposed method.

As an alternative smoothing method, we also propose an estimator:

\[
\hat{\beta}^{\text{HP}} = \arg \min_{\beta} \frac{1}{2} \sum_{i=1}^{n} \int (y_i(x) - \hat{\beta}(x))^2 \, dx + \lambda \sum_{p=1}^{q} \int (\Delta_{p} \beta(x))^2 \, dx.
\]

It corresponds to (9) in time series setting, or, Laplacian regularization (Smola & Kondor, 2003) for univariate data. By the same approximation as the former section, we convert the problem into the optimization problem with objective function:

\[
\frac{1}{2} \sum_{i=1}^{n} \sum_{\ell=1}^{L} (z_{\ell} - b_{\ell})^2 + \lambda \sum_{p=1}^{q} \sum_{\ell=1}^{L} (\Delta_{p} b_{\ell})^2.
\]

For \( \ell = 1, \ldots, L \), the closed form solution is given by

\[
b_{\ell}^{\text{HP}} = \left( I + 2 \lambda \sum_{p=1}^{q} \Delta_{p}^{T} \Delta_{p} \right)^{-1} z_{\ell}.
\]

Consequently, we obtain the estimator \( \hat{\beta}^{\text{HP}}(x) = \sum_{\ell=1}^{L} b_{\ell}^{\text{HP}} \phi_{\ell}(x) \) for \( i = 1, \ldots, n \).

### 3.4 Selection of the number of basis via additional regularization

The methods introduced so far are established by orthonormal basis expansion. What we need to be careful about is the necessity of choosing the number of basis functions in practice because the optimal number of basis functions depends on the complexity of the function. Here, we solve this challenge by constructing the following estimator:

\[
\hat{\beta}^{\text{STF}} = \arg \min_{\beta} \frac{1}{2} \sum_{i=1}^{n} \sum_{\ell=1}^{L} (z_{\ell} - b_{\ell})^2 + \lambda \sum_{i=1}^{n} \sum_{\ell=1}^{L} (\Delta_{p}^{(k)} b_{\ell})^2 + \psi \sum_{\ell=1}^{L} \|b_{\ell}\|^2.
\]
where \( \lambda \) and \( \psi \) are tuning parameters and \( \omega_{\ell} \) is a fixed weight for the \( \ell \)'th coefficients. When we use the principal components as basis functions, we set \( \omega_{\ell} \) as the inverse value of the proportion of variance. The crux of this method is the last \( L \) terms. These are in the form of a group lasso, where the coefficients of each basis function are one group. This makes all the coefficients of the unnecessary basis zero and only the necessary part remains, thus allowing the selection of the number of basis functions. In practice, we select many basis functions beforehand and regard survivors as the essential basis functions. Also, reducing unnecessary principal components promotes smoothing with respect to \( x \)-direction of the function, whereas the middle terms of (13) make the estimator smooth with respect to \( t \)-direction. In what follows, this method is referred to as sparse functional trend filtering.

Regarding the algorithm of this method, the only difference from functional trend filtering is the way \( \{ b_\ell \} \) is updated. Thus, we only describe the updating step of \( \{ b_\ell \} \). An augmented Lagrangian of (13) is

\[
L_t(\{a_t\}, \{b_\ell\}, \{u_t\}) + \psi \sum_{\ell=1}^L \omega_\ell \|b_\ell\|_2^2,
\]

where \( L_t \) is defined in (6). The last \( L \) terms are not smooth with respect to \( \{ b_\ell \} \), but \( L_t \) is differentiable. We then derived the way to update \( \{ b_\ell \} \) by FISTA, and described in Algorithm 4.

---

**Algorithm 4 Update \( b_\ell \) by FISTA.**

1. Initialize \( c_0, s_0 = -1 \) and \( j = 0 \).
2. \( c_{j+1} = 5 \nu_{\ell} \{ \hat{c}_t - V b_t, L_t(c) \} \)
3. \( s_{j+1} = \left[ 1 + \sqrt{1 + 4 \delta_j^2} \right] / 2 \)
4. \( c_{j+1} = c_{j+1} + (s_j - 1)(c_{j+1} - c_j) / s_{j+1} \)
5. If the algorithm has converged, we set \( b_\ell^{(r+1)} = c_{r} \). Otherwise, set \( j = j + 1 \) and go back to Step 2.

---

Regarding the additional tuning parameter \( \psi \), we suggest optimizing \( \psi \) together with \( \lambda \) by the K-fold cross validation given in Algorithm 3.

### 4 SIMULATION STUDIES

In the previous chapters, we develop the two methods. An overview of the simulation is presented in Section 4.1. In Section 4.2, to give a fair comparison of functional trend filtering and other methods, we fix the same number of basis functions for all methods. In Section 4.3, we implement sparse functional trend filtering and advanced functional time series method, and compare their performance. The proposed algorithm is run in R. The average computation time per one-shot is about 1.32 s on a laptop with an Apple M1 processor.

#### 4.1 Procedure

We investigated the performance of the proposed methods together with existing ones through simulation studies. For \( t = 1, \ldots, T (= 50) \) and the domain \( x = [1, 120] \), we adopted the four scenarios of the true trend function:

1. Constant: \( \beta_t(x) = f_1(x) \),
2. Smooth: \( \beta_t(x) = f_2(x) \sin(\frac{\pi x}{14}) \),
3. Piecewise constant: \( \beta_t(x) = \sum_{i=1}^{5} f_i(x) I(10(i-1) < x \leq 10i) \).
4. Varying smoothness: \( \beta_t(x) = f_1(x) + 20 \sin(\frac{\pi x}{2}) + 40 \exp(-30(\frac{\pi x}{2})^2) \),

where \( f_1, \ldots, f_5 \) are sample paths of the Gaussian process associated to RBF kernel \( k(x_1, x_2) = \theta^2 \exp(-\|x_1 - x_2\|^2 / (2\theta^2)) \) with a hyperparameter \( \theta \).

We set \( \theta = 30, 20, 35, 25, 30 \), respectively, for \( f_1, f_2, \ldots, f_5 \). The trends of the functions under the four scenarios are shown in Figure 1. The observed functional data were generated by adding \( N(0, \sigma^2) \) noise at equally spaced \( H = 120 \) points of \( x \), namely, \( x \in \{1, 2, \ldots, H\} \). Remark that we set the noise value so that the SNR would be the same in all the scenarios to fairly compare the competing methods, instead of arbitrarily selecting the noise. Specifically, we set \( \sigma \) to 1/5 (SNR is high), 1/2 (SNR is middle), or 4/5 (SNR is low) of the standard deviation of the signal.

In scenario (1), we examine the abilities of the methods to find the horizontal line in the presence of noise. In scenario (2), we investigate whether the adaptive methods extract the continuous curve from the noisy data. Scenario (3) unearths the capability of the methods to spot the
sharp changes, the points of discontinuities, between intermittent straight horizontal lines. In scenario (4), we test the abilities to catch the trend when the t-direction smoothness of the process varies significantly due to a sharp peak in the middle.

4.2 Studies with fixing basis functions

For the simulated data, we applied the following three methods:

- **FTF**: Functional trend filtering with $k \in \{0, 1, 2\}$.
- **FHP**: Functional HP filter with $k \in \{0, 1, 2\}$.
- **FPC**: The standard functional principal component method using R package “fda.usc.”

FPC is a method proposed by Besse and Ramsay (1986) and widely applied (Aguilera et al., 1999; Erbas et al., 2007; Shang, 2012). Note that we used the estimated principal component functions by FPC as orthonormal functions for FTF and FHP with $L = 5$ (the number of principal functions) to allow comparison independent of basis functions. By 10-fold cross-validation, we select the tuning parameter $\lambda$ from the space $[10^{-3}, 10^3]$ by checking 60 points equally spaced on a logarithmic scale in all scenarios. Furthermore, for each $\lambda$, we set $\rho = \lambda / 10^3$ in Algorithm 2. The motivation for using FHP is to emphasize the importance of sparsity and for employing the FPC method is to convey the extent to which the accuracy is improved by smoothing of basis expansion alone.

The estimated trend functions at $x = 40$ are presented in Figure 2. Based on 150 times repeated simulation, we also report the mean squared error (MSE):

$$
MSE = \frac{1}{T H} \sum_{t=1}^{T} \sum_{x=1}^{H} (\hat{\beta}(t) - \beta(x))^2.
$$
Table 1, where \( \hat{\beta}_1(x) \) is the estimated function. Overall, the proposed FTF tended to perform better than the other methods. Further, we can see from Figure 2 that FPC provided undersmoothed trend estimate compared with FTF and FHP, which is related to the overall performance in terms of MSE reported in Table 1.

Interestingly, the performance of FTF and FHP were quite different, although the only methodological difference is whether \( L^2 \)-norm or squared \( L^2 \)-norm is adopted in the penalty. For example, in Scenario 3, the performance of FHP was almost the same as that of FPC, while FTF provided better results. This is attributed to the fact that FHP does not produce sparsity. Regarding the performance of FTF depending on \( k \), it is observed that FTF with \( k = 0 \) provided the most accurate results in Scenario 3 because the true trend admits a piecewise constant structure that
FTF with \( k = 0 \) is considered to work well. The importance of FTF with \( k = 0 \) is similar in Scenario 1, where it also enjoys sparsity. In the other scenarios, however, the piecewise constant structure seems rather limited, and the performance of FTF with \( k = 1,2 \) is more appealing. It is worth noting that FTF with \( k = 1 \) performed well in Scenario 4. Around the peak of Scenario 4, the smoothness of the trend changes abruptly. The change in smoothness is nearly equal to the change in the amount of difference. Hence, the sharp peak of Scenario 4 is the point where the supremacy of FTF exists. By contrast, in Scenario 2, FTF is slightly inferior to FHP. One possible reason is that FTF is a numerical solution obtained by iterative approximation while FHP is an analytical solution.

### 4.3 Studies with selecting the number of basis functions

The above simulation showed that FTF accurately estimated the trend even with sudden changes. However, we need to choose the appropriate number of basis functions. Hence, we applied SFTF, introduced in Section 3.4, and investigated whether the number of basis functions could be properly selected. Specifically, we set \( L = 10 \) first, and then applied SFTF and cut off unnecessary basis functions. We implemented 150 simulations and calculated MSE of SFTF and mean of the number of basis functions. We searched for the optimal values of \( (\lambda, \psi) \) from the space \([10^{-3}, 10^3] \times [10^{-3}, 10^3]\) by checking \( 60 \times 20 \) points equally spaced on a logarithmic scale in all scenarios. To verify whether SFTF is superior to the existing functional time series method, we implemented dynamic functional principal component analysis (DFPC), which incorporates serial dependence (Aue et al., 2015; Hörmann et al., 2015). The R package “freqdom.fda” does not mention anything about parameter selection. Then, we set the parameters to minimize the MSE. Namely, we compare SFTF to DFPC with oracle parameters.

Table 2 presents the MSE of DFPC and that of SFTF. We present three considerations here. First, we chose parameters that favored DFPC, but SFTF dominated it. Hence, the superiority of SFTF, or the locally adaptive method, is solidified. Second, the number of basis functions used

| Scenario | SNR     | SFTF 0 | SFTF 1 | SFTF 2 | DFPC  |
|----------|---------|--------|--------|--------|-------|
| 1        | High    | 0.388  | 0.388  | 0.388  | 3.259 |
|          | Middle  | 2.424  | 2.424  | 2.433  | 20.353|
|          | Low     | 6.204  | 6.241  | 6.509  | 52.101|
| 2        | High    | 1.588  | 1.430  | 1.438  | 3.518 |
|          | Middle  | 9.755  | 8.575  | 8.392  | 21.620|
|          | Low     | 23.914 | 21.854 | 21.379 | 55.238|
| 3        | High    | 1.899  | 1.830  | 1.781  | 2.867 |
|          | Middle  | 9.108  | 10.095 | 10.223 | 17.903|
|          | Low     | 22.210 | 24.525 | 25.377 | 45.826|
| 4        | High    | 1.555  | 1.152  | 1.493  | 5.332 |
|          | Middle  | 9.413  | 9.091  | 9.188  | 33.296|
|          | Low     | 34.104 | 29.557 | 25.465 | 85.229|
by the SFTF depends on the SNR. The result tends to incorporate more basis functions, the larger the noise. This intimates that denoise by trend filtering can be more potent than selecting useful basis functions. Third, the performance of SFTF reported in Table 2 is superior to that of FTF with the fixed number of basis functions as reported in Table 1. The primary reason is that the number of basis functions (i.e., information when applying FTF) is optimized. By cutting down on the number of basis functions when there is a lot of unnecessary noise and by importing many basis functions when much information is needed, the estimation can be improved by the amount of selection. This is not accomplished by predetermining the basis functions or by reserving more. Thus, choosing the number of basis functions by excluding redundant ones plays a critical role in increasing the accuracy.

5 | APPLICATIONS

5.1 | Australian fertility rates

Fertility rates in Australia have been declining seriously as in other developed countries. We examined the data “Australiasmoothfertility,” which is available from R package “rainbow.” The original data, obtained from the Australian Bureau of Statistics, describes the age-specific number of live births per 1000 females of ages 15, 16, ..., 49 from 1921 to 2015. The data are functional data, and each function represents the age-specific number between 15 and 49 in a year. Figure 3 shows the curves with rainbow colors. The colors indicate that the oldest curve is red, the newest curve is purple and the others are colored in the same order as a rainbow.

Here we applied SFTF to the dataset and set \( L = 10 \) first. We selected tuning parameters \((\lambda, \psi)\) from the space \([10^{-2}, 10^2] \times [10^{-3}, 10^0]\) by checking 40 \( \times 10 \) points equally spaced on a logarithmic scale in all scenarios. The number of principal components selected by SFTF was five.

Figure 4 shows the number of births per 1000 females of ages 20 and 30 in all years and curves fitted by FPC and SFTF. Figure 5 displays absolute values of first-order differences and second-order differences in scores of the first principal component (PC1) between the years and their trend filtered versions. First, compared with FPC, the ability of SFTF to serve as a smoother is confirmed in Figure 4. It eliminates small noises but retains the significant change points.

Next, in common between ages 20 and 30 in Figure 4, we find abrupt changes in 1961 and 1972. After World War II, the fertility rate had increased until 1961, although the first oral contraceptive pill was released in Australia in 1961. Furthermore, in 1972, the prime minister of Australia at that time abolished the 27.5% luxury tax on all contraceptives (McLennan, 1998). It increased the use of the pills especially among young people, and the trends are reflected as the sharp change points in plots in Figure 4. Moreover, from the upper right plot in Figure 5, the structure in the sense of second-order difference is considered to change at 40th and 50th points; namely, large structural changes occurred from 1960 to 1962 and from 1970 to 1972. The lower left plot suggests, in terms of second-order difference, the structure changes at 24th point (i.e., around 1945), implying that the trend of the fertility rate changed after the end of World War II. Owing to the sparsity in differences in trend, we easily detect those underlying events. In addition, because the detected points from the plots tend to be overlapped between \( k = 1 \) and 2 in

![Figure 3](image-url)  
**FIGURE 3**  Age-specific Australian fertility rates curves for ages 15 to 49 observed from 1921 to 2015 (in the same order as the color in a rainbow)
Figure 4. At age 20 and 30, the number of births per 1000 females (round points), trends fitted by FPC (red lines in left column), and trends fitted by first-order SFTF (blue lines in right column).

Figure 5. Top left: Absolute values of first-order differences in scores of PC1. Top right: Absolute values of first-order differences in the trend filtered scores. Bottom left: Absolute values of second-order differences in the scores. Bottom right: Absolute values of second-order differences in the trend filtered scores.

Figure 5, trend filtering would be able to stably extract the turning points regardless of the order $k$. By contrast, we hardly find the structural properties of data from the original scores of the principal component. Therefore, the result proves the ability of trend filtering to capture sharp changes.
Infection with the novel COVID-19 has been spreading since 2020 and has brought about many deaths worldwide. Thus, analyzing the situation becomes increasingly important. For instance, Tang et al. (2020) exploited some functional time series methods to analyze the COVID-19 data in the United States. In this study, we investigate the number of people infected by COVID-19 by prefecture in Japan, which is available at https://www3.nhk.or.jp/news/special/coronavirus/data-widget/, and we scale the number by population of each prefecture available at https://www.stat.go.jp/data/nihon/02.html. We handled the number of infected people per million in each prefecture from January 16, 2020, to March 9, 2021. Each prefecture is treated as a vertex on a graph, and if the prefectures are adjacent to each other, they are assumed to be connected by an edge. We regard the number of COVID-19 cases over the period as a function at each vertex on a graph; namely, $y_i(x)$ in (11) represents COVID-19 cases on $x$th day in $i$th prefecture.

We investigated the relationship between prefectures in terms of infection. We focus on the figure with the date fixed. The observed data on 395th day are shown in the upper left panel in Figure 6. In this panel, the coloring is not regularly distributed. Because the infections should be somewhat smooth in distribution except for the singularities, we smoothed this observation in a spatial sense. We applied FPC and second-order FTF. We plot the data fitted by FPC in the upper right panel and second-order FTF in the lower left panel, where the value of $\lambda$ was selected as the argument of the minimum MSE from $[10^{-3}, 10^{3}]$. We also applied FTF with $k = 1$, but the result is almost the same; thus, we do not display it here.

We focus our discussion on a qualitative visual analysis of the results. The result of FPC remained jagged and almost unchanged from the real data. In contrast, after FTF was applied, the subtle variations were significantly denoised. This is an indication of the smoothing feature of the FTF. More noteworthy is that FTF spotted an outstanding (dark-colored) prefecture, Tokyo. FTF kept this segment unchanged from the real data.

![Figure 6](image-url)  
**Figure 6**  
Top left: The observed number of infected people by prefecture on day 395. Top right: The number of infected people by prefecture on day 395, smoothed by FPC. Bottom left: The number of infected people by prefecture on day 395, smoothed by second-order FTF.
Smoothing and spotting large points are seemingly contradictory, but the excel of FTF is that it makes both possible. Evidently, FTF is able to localize its estimates around strong inhomogeneous spikes, which implies that it is able to detect the event or spot of interest.

6 | DISCUSSION

In this paper, we proposed a functional version of the locally adaptive smoothing technique known as trend filtering for smoothing functional time series and spatial data. The need to consider group lasso + fused lasso like penalty does not allow for a trivial extension of the scalar version, but we developed an efficient optimization algorithm to obtain trend estimation and discussed the choice of the tuning parameter. Through simulation and empirical studies, we demonstrated that the proposed method can identify abrupt changes and is more accurate than existing methods.

Moreover, in time series data, we can select the number of basis functions by adding a penalty. The reduction of unnecessary basis functions denoises the functions themselves, whereas trend filtering is smoother with respect to time direction. As a result, the performance of the simulation is also improved, showing that choosing the number of basis functions led to more accurate estimates than just taking more basis functions. On the whole, penalties are the key to the methods we developed.

The optimization problem for computing the proposed method can be regarded as a generalization and a combination of grouped and fused lasso estimation. Thus, it would be interesting to apply the proposed optimization techniques to other statistical problems, for example, regression analysis with complicated sparsity-inducing penalty functions.

ACKNOWLEDGMENTS

Research of the authors was supported in part by JSPS KAKENHI Grant Numbers 18H03628, 20H00080, and 21H00699 from Japan Society for the Promotion of Science.

ORCID

Tomoya Wakayama https://orcid.org/0009-0007-2611-9535

REFERENCES

Aguilera, A. M., Ocaña, F. A., & Valderrama, M. J. (1999). Forecasting time series by functional PCA. discussion of several weighted approaches. Computational Statistics, 14(1), 443–467.

Alaíz, C. M., Barbero, A., & Dorronsoro, J. R. (2013). Group fused lasso. In International Conference on Artificial Neural Networks, Springer, pp. 66–73.

Aue, A., Horváth, L., & Pellatt, D. (2017). Functional generalized autoregressive conditional heteroskedasticity. Journal of Time Series Analysis, 38(1), 3–21.

Aue, A., & Klepsch, J. (2017). Estimating functional time series by moving average model fitting. arXiv preprint arXiv:1701.00770.

Aue, A., Norinho, D. D., & Hörmann, S. (2015). On the prediction of stationary functional time series. Journal of the American Statistical Association, 110(509), 378–392.

Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences, 2(1), 183–202.

Besse, P., & Ramsay, J. O. (1986). Principal components analysis of sampled functions. Psychometrika, 51(2), 285–311.

Besse, P. C., Cardot, H., & Stephenson, D. B. (2000). Autoregressive forecasting of some functional climatic variations. Scandinavian Journal of Statistics, 27(4), 673–687.

Bosq, D. (2000). Linear processes in function spaces: Theory and applications, Vol. 149. Springer Science & Business Media.

Boyd, S., Parikh, N., & Chu, E. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc.

Caballero, W., Giraldo, R., & Mateu, J. (2013). A universal kriging approach for spatial functional data. Stochastic Environmental Research and Risk Assessment, 27(7), 1553–1563.

Cerovecki, C., Franço, C., Hörmann, S., & Zakoian, J.-M. (2019). Functional garch models: The quasi-likelihood approach and its applications. Journal of Econometrics, 209(2), 353–375.

Erbaş, B., Hyndman, R. J., & Gertig, D. M. (2007). Forecasting age-specific breast cancer mortality using functional data models. Statistics in Medicine, 26(2), 458–470.

Gao, Y., Shang, H. L., & Yang, Y. (2019). High-dimensional functional time series forecasting: An application to age-specific mortality rates. Journal of Multivariate Analysis, 170, 232–243.

Giraldo, R., Delicado, P., & Mateu, J. (2011). Ordinary kriging for function-valued spatial data. Environmental and Ecological Statistics, 18(3), 411–426.

Hodrick, R. J., & Prescott, E. C. (1997). Postwar U.S. business cycles: An empirical investigation. Journal of Money, Credit and Banking, 29(1), 1–16. http://www.jstor.org/stable/2935682

Hörmann, S., Horváth, L., & Reeder, R. (2013). A functional version of the arch model. Econometric Theory, 29(2), 267–288.

Hörmann, S., Kidziński, L., & Hallin, M. (2015). Dynamic functional principal components. Journal of the Royal Statistical Society: Series B: Statistical Methodology, 77(2), 319–348.

Horváth, L., & Kokoszka, P. (2012). Inference for functional data with applications, Vol. 200. Springer Science & Business Media.

Kim, S.-J., Koh, K., Boyd, S., & Gorinevsky, D. (2009). ρ trend filtering. SIAM Review, 51(2), 339–360.

Klepsch, J., & Klüppelberg, C. (2017). An innovations algorithm for the prediction of functional linear processes. Journal of Multivariate Analysis, 155, 252–271.
Klepsch, J., Klüppelberg, C., & Wei, T. (2017). Prediction of functional arma processes with an application to traffic data. *Econometrics and Statistics*, 1, 128–149.

Kokoszka, P., & Reimherr, M. (2017). *Introduction to functional data analysis*. CRC Press.

Kühnert, S. (2020). Functional arch and garch models: A yule-walker approach. *Electronic Journal of Statistics*, 14(2), 4321–4360.

Lounici, K., Pontil, M., Van De Geer, S., & Tsybakov, A. B. (2011). Oracle inequalities and optimal inference under group sparsity. *Annals of Statistics*, 39(4), 2164–2204.

Mammen, E., & van de Geer, S. (1997). Locally adaptive regression splines. *The Annals of Statistics*, 25(1), 387–413.

McLennan, W. (1998). Australian social trends 1998.

Menafoglio, A., Grujic, O., & Caers, J. (2016). Universal kriging of functional data: Trace-variography vs cross-variography? Application to gas forecasting in unconventional shales. *Spatial Statistics*, 15, 39–55.

Menafoglio, A., Secchi, P., & Dalla Rosa, M. (2013). A universal kriging predictor for spatially dependent functional data of a hilbert space. *Electronic Journal of Statistics*, 7, 2209–2240.

Nerini, D., Monestiez, P., & Manté, C. (2010). Cokriging for spatial functional data. *Journal of Multivariate Analysis*, 101(2), 409–418.

Ramdas, A., & Tibshirani, R. J. (2016). Fast and flexible admm algorithms for trend filtering. *Journal of Computational and Graphical Statistics*, 25(3), 839–858.

Ramsay, J. O. (2004). Functional data analysis. *Encyclopedia of Statistical Sciences*, 4.

Ramdas, A. (2020). Point and interval forecasts of age-specific fertility rates: A comparison of functional principal component methods. *Journal of Population Research*, 29(3), 249–267.

Smola, A. J., & Kondor, R. (2003). Kernels and regularization on graphs, *Learning theory and kernel machines*. Springer, pp. 144–158.

Spangenberg, F. (2013). Strictly stationary solutions of arma equations in banach spaces. *Journal of Multivariate Analysis*, 121, 127–138.

Steidl, G., Didas, S., & Neumann, J. (2006). Splines in higher order tv regularization. *International Journal of Computer Vision*, 70(3), 241–255.

Tang, C., Wang, T., & Zhang, P. (2020). Functional data analysis: An application to covid-19 data in the united states. arXiv preprint arXiv:2009.08363.

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

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1), 91–108.

Tibshirani, R. J. (2014). Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1), 285–323.

van Delft, A., & Eichler, M. (2018). Locally stationary functional time series. *Electronic Journal of Statistics*, 12(1), 107–170.

Wang, Y.-X., Sharpnack, J., Smola, A. J., & Tibshirani, R. J. (2016). Trend filtering on graphs. *The Journal of Machine Learning Research*, 17(1), 3651–3691.

Wang, Y.-X., Smola, A., & Tibshirani, R. (2014). The falling factorial basis and its statistical applications. In *International Conference on Machine Learning*, PMLR, pp. 730–738.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1), 49–67.

---

**How to cite this article:** Wakayama, T., & Sugasawa, S. (2023). Trend filtering for functional data. Stat, 12(1), e590. [https://doi.org/10.1002/sta4.590](https://doi.org/10.1002/sta4.590)