Drift Removal for Time Series Data Using Quantile Trend Filtering

Halley Brantley* Joseph Guinness† and Eric C. Chi‡

Abstract

Abstract

Keywords: Key words

*Department of Statistics, North Carolina State University, Raleigh, NC 27695 (E-mail: hlbrantl@ncsu.edu)

†Department of Statistics, North Carolina State University, Raleigh, NC 27695-8203 (E-mail: jsguinne@ncsu.edu)

‡Department of Statistics, North Carolina State University, Raleigh, NC 27695-8203 (E-mail: eric_chi@ncsu.edu).

1 Introduction

1.1 Background

- Oh et al. (2011) unified framework for non-parametric quantile regression by approximating check loss function with quadratic loss near zero.
- Koenker et al. (1994) introduce quantile smoothing splines. We need to discuss how our approach is different.
- Kim et al. (2009) introduce the concept of ℓ_1-trend filtering.
- Tibshirani (2014) describes properties of trend filtering using quadratic loss. Shows that trend filtering estimates adapt to the local level of smoothness much better than smoothing splines, and exhibit a remarkable similarity to locally adaptive regression splines. Prove that (with the right choice of tuning parameter) the trend filtering estimate converges to the true underlying function at the minimax rate for functions whose kth derivative is of bounded variation.
- Ning et al. (2014) address problem of estimating a smooth baseline in noisy data with drift.
- Takeuchi et al. (2006) Nonparametric quantile regression using SVM with Gaussian RBF kernels and check (pinball) loss.
- Yuan (2006) Comparison of cross-validation methods for quantile smoothing splines.

We propose to use the trend filtering penalty with the check loss function to produce a non-parametric quantile regression estimate that can be computes using a linear time algorithm for removing trends in time series. The formulation was proposed by Kim et al. (2009) as a possible extension of ℓ_1-trend filtering but not studied. Moreover we extend the basic framework to model multiple quantiles and ensure non-crossing.

1.2 Application

Examples:

Figure 1: Raw Data - three collocated sensors

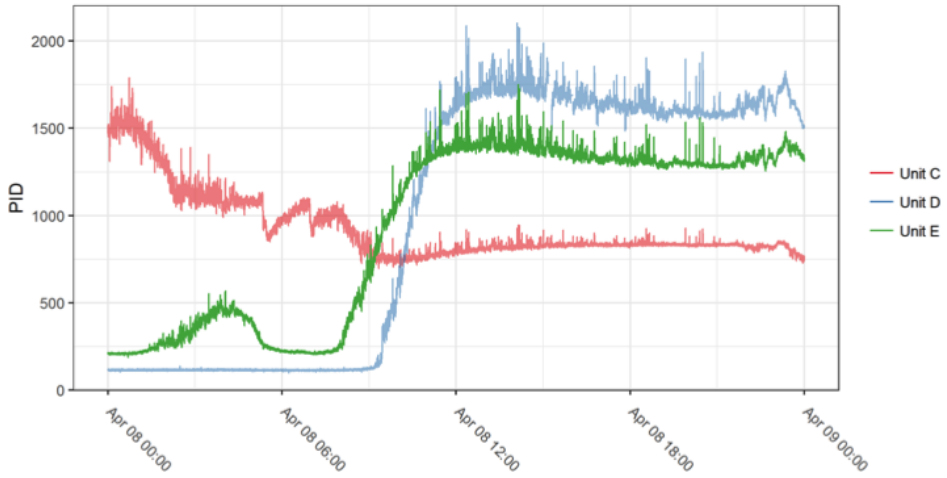
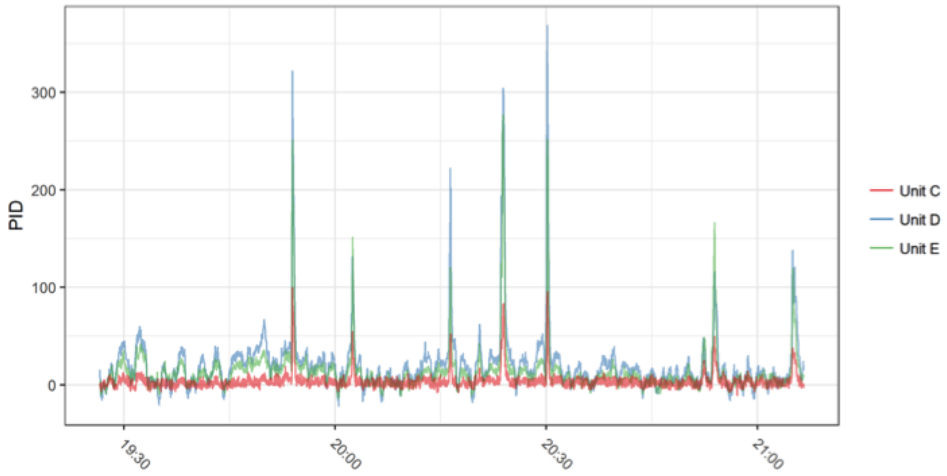


Figure 2: Detrended data - three collocated sensors



- Air quality (Apte et al., 2017)
- ECG (Sanyal et al., 2012)
- Chromatogram baseline estimation (Ning et al., 2014; Ilewicz et al., 2015)
- Galaxy spectrum baseline estimation (Ilewicz et al., 2015; Bacher et al., 2016)
- Identifying absorption dips from black body radiation

Need to decide between detrending versus detrending + denoising. The BEADS (Ning et al., 2014) method does both. We may wish to focus on detrending and then use wavelet

SURE denoising as a postprocessing step, i.e. do a two-stage procedure, both of which can be done in linear time. On paper this should be faster than the BEADS procedure. Or we may just want to stick with detrending. There's Matlab code for BEADS, and the BEADS paper also points to two other popular methods in chromotography.

Things to do:

- Convergence of the algorithm
- Convergence rate (He and Yuan, 2012, 2015; Davis, 2017)
- Timing experiments of LP versus Spingarn
- Make an R package - detrendr
- Compare on synthetic data quality of solution with existing methods
- Do comparisons on real data examples

2 Quantile Regression

The classic least squares regression is notoriously sensitive to outliers. One remedy to blunt the influence of outliers is to compute the least absolute deviations (LAD) solution in place of the least squares one. Given a design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ and continuous responses $\mathbf{y} \in \mathbb{R}^n$, we estimate a regression vector $\boldsymbol{\theta} \in \mathbb{R}^p$ so that $\mathbf{X}\boldsymbol{\theta}$ is a good approximation of $\mathbf{y}$. The LAD estimator is a solution to the problem

$$\min_{\boldsymbol{\theta}} \frac{1}{n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_1.$$

The above optimization problem generalizes the notion of the median of a collection of numbers. A median μ of n reals $y_1, \ldots, y_n$ is the minimizer of the function

$$f(u) \quad = \quad \frac{1}{n} \sum_{i=1}^{n} |y_i - \theta|.$$

Recall that the median is the 50th percentile or 0.5-quantile, namely half of the y_i are less than or equal to μ and the other half is greater than or equal to μ. The median can be

generalized to arbitrary τ-quantiles for $\tau \in (0,1)$ to give us quantile regression (Koenker and Bassett, 1978).

First define the so-called "check function"

$$\rho_\tau(\Delta) \quad = \quad \begin{cases} \tau\Delta & \Delta \geq 0 \\ -(1-\tau)\Delta & \Delta < 0. \end{cases}$$

Then the τth quantile of the y_i is a minimizer of the function

$$f_\tau(\theta) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_i - \theta).$$

Returning to the regression context, we can generalize LAD regression to quantile regression, namely computing the minimizer of the function

$$f_\tau(\boldsymbol{\theta}) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_i - \langle \mathbf{x}_i \mid \boldsymbol{\theta} \rangle),$$

where $\mathbf{x}_i \in \mathbb{R}^p$ denotes the ith row of $\mathbf{X}$.

3 Trend Filtering

In the trend filtering problem (Kim et al., 2009; Tibshirani, 2014), one is interested in finding an adaptive polynomial approximation to noisy data $\mathbf{y} \in \mathbb{R}^n$ by solving the following convex problem.

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{2n}\|\mathbf{y} - \boldsymbol{\theta}\|_2^2 + \lambda \|\mathbf{D}^{(k+1)}\boldsymbol{\theta}\|_1,$$

where $\lambda \geq 0$ is a regularization parameter that trades off the emphasis on the data fidelity term and the matrix $\mathbf{D}^{(k+1)} \in \mathbb{R}^{(n-k-1)\times n}$ is the discrete difference operator of order $k+1$. To understand the purpose of penalizing $\mathbf{D}^{(k+1)}$ consider the difference operator when $k = 0$.

$$\mathbf{D}^{(1)} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix}$$

Thus, $\|\mathbf{D}^{(1)}\boldsymbol{\theta}\|_1 = \sum_{i=1}^{n-1}|\theta_i - \theta_{i+1}|$ which is just total variation denoising in one dimension. The penalty incentivizes solutions which are piece-wise constant. For $k \geq 1$, the difference operator $\mathbf{D}^{(k+1)} \in \mathbb{R}^{(n-k-1)\times n}$ is defined recursively as follows

$$\mathbf{D}^{(k+1)} = \mathbf{D}^{(1)}\mathbf{D}^{(k)}.$$

By penalizing the $k+1$ fold composition of the discrete difference operator, we obtain solutions which are piecewise polynomials of order k.

4 Quantile Trend Filtering

We combine the ideas of quantile regression and trend filtering, namely consider the signal approximation problem, where the design $\mathbf{X}$ is the identity matrix.

The estimation of the quantile trend filtering model can be posed as the following optimization problem.

$$\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n}\rho_\tau(y_i - \theta_i) + \lambda\|\mathbf{D}^{(k)}\boldsymbol{\theta}\|_1, \tag{4.1}$$

where λ is a nonnegative tuning parameter. As with the classic quantile regression, the quantile trend filtering problem can be solved by a linear program. We argue that it is better solved by Spingarn's method of partial inverses.

5 Related Work

- Quantile splines (Oh et al., 2011)
- BEADS

6 Spingarn's method of partial inverses

We first review Spingarn's method (Spingarn, 1985), which solves the following equality constrained convex problem:

$$\begin{aligned} &\text{minimize } \psi(\mathbf{x}) \\ &\text{subject to } \mathbf{x} \in V, \end{aligned} \tag{6.1}$$

where V is a subspace. The problem (6.1) can be expressed as the unconstrained optimization problem

$$\text{minimize } \psi(\mathbf{x}) + \iota_V(\mathbf{x}), \tag{6.2}$$

where ι_V is the indicator function of the set V. Spingarn's method applies Douglas-Rachford splitting to the problem (6.2) to give the following updates.

$$\begin{aligned}
\mathbf{x}^{(k+1)} &= \text{prox}_{t\psi}(\mathbf{z}^{(k)}) \\
\mathbf{y}^{(k+1)} &= P_V\left(2\mathbf{x}^{(k+1)} - \mathbf{z}^{(k)}\right) \\
\mathbf{z}^{(k+1)} &= \mathbf{z}^{(k)} + \lambda^{(k)}\left(\mathbf{y}^{(k+1)} - \mathbf{x}^{(k+1)}\right).
\end{aligned}$$

The parameter t is a step-size and $\lambda^{(k)}$ is Krasnosel'skiĭ-Mann iteration (Need citation). We require that $\lambda^{(k)} \in]0, 2[$ and that $\sum_n \lambda^{(k)}(2-\lambda^{(k)}) = \infty$. The mapping P_V is the orthogonal projection onto the set V. Note that the algorithm iterates $\mathbf{x}^{(k)}$ will converge to a solution to problem (6.1) (Combettes and Wajs, 2005). We need to experiment with different over / under-relaxation parameters $\lambda^{(k)}$ and step sizes t. It will converge if we take $\lambda^{(k)} = 1$ and $t = 1$, but we may be able to converge faster in practice by taking non-trivial values.

7 Applying Spingarn's Method to Quantile Trend Filtering

To simplify the notation we suppress the order k and write $\mathbf{D}^{(k)}$ as $\mathbf{D}$. We can reformulate our optimization problem (4.1) as the following equality constrained convex optimization problem.

$$\begin{aligned}
\text{minimize} \quad & f_1(\boldsymbol{\theta}) + f_2(\boldsymbol{\eta}) \\
\text{subject to} \quad & \boldsymbol{\eta} = \mathbf{D}\boldsymbol{\theta}
\end{aligned}$$

where

$$f_1(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_i - \theta_i) \qquad \text{and} \qquad f_2(\boldsymbol{\eta}) = \lambda\|\boldsymbol{\eta}\|_1.$$

If we set $\psi(\boldsymbol{\theta}, \boldsymbol{\eta}) = f_1(\boldsymbol{\theta}) + f_2(\boldsymbol{\eta})$ and $V = \{\mathbf{z}^\mathsf{T} = (\boldsymbol{\theta}^\mathsf{T}, \boldsymbol{\eta}^\mathsf{T}) : \boldsymbol{\eta} = \mathbf{D}\boldsymbol{\theta}\}$, then we can apply Spingarn's method. Note that

$$\begin{aligned} \text{prox}_{th}(\boldsymbol{\theta}, \boldsymbol{\eta}) &= (\text{prox}_{tf_1}(\boldsymbol{\theta}), \text{prox}_{tf_2}(\boldsymbol{\eta})) \\ P_V(\boldsymbol{\theta}, \boldsymbol{\eta}) &= \begin{pmatrix} \mathbf{I} \\ \mathbf{D} \end{pmatrix} \left(\mathbf{I} + \mathbf{D}^\mathsf{T}\mathbf{D}\right)^{-1} \left(\boldsymbol{\theta} + \mathbf{D}^\mathsf{T}\boldsymbol{\eta}\right), \end{aligned}$$

where the projection P_V requires a banded linear system solve, with bandwidth $k+1$. This linear solve can be accomplished in $\mathcal{O}(n(k+1)^2)$. The first solve using a banded Cholesky decomposition requires $\mathcal{O}(n(k+1)^2)$. Subsequent solves require $\mathcal{O}(n(k+1))$. We can use RccpArmadillo to do banded Cholesky (Eddelbuettel and Sanderson, 2014).

Proximal mappings

We need the proximal mappings for tf_1 and tf_2.

$$\begin{aligned} \left[\text{prox}_{tf_1}(\boldsymbol{\theta})\right]_i &= y_i - \text{prox}_{(t/n)\rho_\tau}(y_i - \theta_i), \\ \left[\text{prox}_{tf_2}(\boldsymbol{\eta})\right]_j &= S(\eta_j, t\lambda). \end{aligned}$$

The proximal mapping for tf_2 is the element-wise softthresholding operator. We now derive the proximal mapping of $\rho_\tau(\Delta)$, which can be evaluated in closed form. We need to find the minimizer of the following univariate function

$$g_\tau(\Delta) = \Delta[\tau - I(\Delta < 0)] + \frac{n}{2t}(\Delta - w)^2,$$

where $w \in \mathbb{R}$ is given and $I(\Delta < 0)$ is 0 when $\Delta < 0$ and 1 otherwise.

The subgradient of $\rho_\tau(\Delta) = \Delta[\tau - I(\Delta < 0)]$ is given by

$$\partial\rho_\tau(\Delta) = \begin{cases} \tau & \text{if } \Delta > 0 \\ \tau - 1 & \text{if } \Delta < 0 \\ [\tau - 1, \tau] & \text{if } \Delta = 0. \end{cases}$$

The stationary condition is

$$\frac{n}{t}[w - \Delta] \quad \in \quad \partial\rho_\tau(\Delta).$$

Therefore, the proximal mapping is given by

$$\text{prox}_{(t/n)\rho_\tau}(w) \quad = \quad \begin{cases} w - \tau\frac{t}{n} & \text{if } w > \tau\frac{t}{n} \\ w + (1-\tau)\frac{t}{n} & \text{if } w < -(1-\tau)\frac{t}{n} \\ 0 & \text{if } -(1-\tau)\frac{t}{n} \leq w \leq \tau\frac{t}{n}. \end{cases}$$

Computational Costs

Precomputation

The following calculations need only be done once.

- $\mathcal{O}(n(k+1)^2)$ to compute the banded Cholesky factorization of $\mathbf{I} + [\mathbf{D}^{(k)}]^\mathsf{T}[\mathbf{D}^{(k)}]$

Per-Iteration

The following calculations will be done every iteration.

- $\mathcal{O}(n)$ to compute $\text{prox}_{t\psi}(\boldsymbol{\theta}, \boldsymbol{\eta})$
- $\mathcal{O}((k+1)(n-k+1))$ to compute $\boldsymbol{\theta} + [\mathbf{D}^{(k)}]^\mathsf{T}\boldsymbol{\eta}$
- $\mathcal{O}(n(k+1))$ to compute $\boldsymbol{\phi} = (\mathbf{I} + [\mathbf{D}^{(k)}]^\mathsf{T}[\mathbf{D}^{(k)}])^{-1}(\boldsymbol{\theta} + \mathbf{D}^\mathsf{T}\boldsymbol{\eta})$
- $\mathcal{O}((k+1)(n-k+1))$ to compute $\begin{pmatrix} \mathbf{I} \\ \mathbf{D}^{(k)} \end{pmatrix} \boldsymbol{\phi}$

  The total cost is $\mathcal{O}(nk)$.

7.1 Summary

- The overall computational complexity is essentially linear $\mathcal{O}(nk^2)$ for the initial banded Cholesky decomposition and the per-iteration complexity is $\mathcal{O}(nk)$.
- One could also apply Anderson acceleration (Walker and Ni, 2011) to reduce the number of Spingarn updates, since the Douglas-Rachford algorithm is a fixed point algorithm.

8 Applying Spingarn's Method to Quantile Trend Filtering version 2

Following the specialized ADMM algorithm for trend filtering (Ramdas and Tibshirani, 2016), we can also take advantage of fast exact solvers of the one-dimensional fussed lasso problem (Davies and Kovac, 2001; Johnson, 2013). The first method is based on taut strings, and the second is based on dynamic programming. Both of these methods run in linear time. A third exact method with worst case quadratic penalty but linear time in typical cases was proposed by Condat (2013). C code is available for all three methods. We should compare them.

What is the reparameterization?

$$\begin{aligned} \text{minimize} \quad & f_1(\boldsymbol{\theta}) + f_2(\boldsymbol{\eta}) \\ \text{subject to} \quad & \boldsymbol{\eta} = \mathbf{D}\boldsymbol{\theta} \end{aligned}$$

where

$$f_1(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_i - \theta_i) \qquad \text{and} \qquad f_2(\boldsymbol{\eta}) = \lambda \|\mathbf{D}^{(1)}\boldsymbol{\eta}\|_1.$$

Difference with version 1: If we are interested in a 3rd order penalization, then $\mathbf{D} = \mathbf{D}^{(2)}$ in version 2 as opposed to $\mathbf{D} = \mathbf{D}^{(3)}$ in version 1. The proximal mapping for f_2 is solved using one of the exact solvers (Davies and Kovac, 2001; Johnson, 2013; Condat, 2013). Show some timing results between the different versions. Point readers to Ramdas and Tibshirani (2016) for a discussion of why this minor reparameterization may lead to speed up.

9 Other Practical Issues

- Read compressed data
- May want to read data directly in C and not pull into R; make R just an interface
- Use historical data to choose τ and λ

10 To Do

- Find out if we can use the EPA data. Add co-authors?
- Make version 2 with the three fast fused lasso solvers
- Add homotopy / warm start
- Add model selection
- Do experiments to evaluate different choices of t and $\lambda^{(k)}$
- See if Anderson acceleration helps. Talk to Tim Kelley?
- Write vignette
- Do comparisons with BEAD
- Do wind polar plots with and without removing trends
- Compare with Splines

11 Numerical Studies

We compare our detrending method with BEADS and the more general nonparametric quantile method introduced by Oh et al. (2011).

12 ADMM for Smoothing Windows

To our knowledge, no one has addressed the problem of finding smooth quantile trends of series that are too large to be processed simultaneously. We propose an alternating direction method of multipliers (ADMM) algorithm for solving large problems in a piecewise fashion. The ADMM algorithm, described by Boyd et al. (2011), relies on the idea of dual ascent. If we consider the equality optimization problem

$$\text{minimize } f(x) \tag{12.1}$$

$$\text{subject to } Ax = b \tag{12.2}$$

with variable $x \in \mathbf{R}^n$, where $A \in \mathbf{R}^{mxn}$ and $f : \mathbf{R}^n \to \mathbf{R}$ is convex. The corresponding Lagrangian function is

$$L(x, \omega) = f(x) + \omega^T(Ax - b) \tag{12.3}$$

and the corresponding dual problem is

$$\text{maximize } \inf_x L(x, \omega) \tag{12.4}$$

Assuming that strong duality holds (Slater's constraint: there exists a solution in the interior of the domain), the optimal values of the primal and dual variables are the same. The dual ascent method, uses gradient ascent to solve the dual problem and thus the primal problem using the iterates

$$x^{k+1} = \arg\min L(x, \omega^k) \tag{12.5}$$

$$\omega^{k+1} = \omega^k + \alpha^k(Ax^{k+1} - b) \tag{12.6}$$

where $\alpha^k > 0$ is a step size and k is the iteration counter. In many cases (including our problem), the x update fails because the problem is unbounded below. To remedy this issue the augmented Lagrangian was introduced:

$$L_\gamma(x, \omega) = f(x) + \omega^T(Ax - b) + \frac{\gamma}{2}||Ax - b||_2^2 \tag{12.7}$$

where $\gamma > 0$ is called the penalty parameter. Applying dual ascent to the modified problem results in the method of multipliers algorithm which converges under more general conditions:

$$x^{k+1} = \arg\min L_\gamma(x, \omega^k) \tag{12.8}$$

$$\omega^{k+1} = \omega^k + \gamma(Ax^{k+1} - b) \tag{12.9}$$

Consider a new optimization problem with a separable objective function.

$$\text{minimize } f_1(x_1) + f_2(x_2) \tag{12.10}$$

$$\text{subject to } A_1x_1 + A_2x_2 = b \tag{12.11}$$

where $x = (x_1, x_2)$, $i \in \{1, 2\}$, and $x_i \in \mathbf{R}^{n_i}$. The dual ascent updates can be expressed as

$$x_i^{k+1} = \arg\min L_i(x_i, \omega^k) \tag{12.12}$$

$$\omega^{k+1} = \omega^k + \alpha^k(Ax^{k+1} - b) \tag{12.13}$$

however we lose the separability if the augmented Lagrangian is used instead, i.e. the augmented Lagrangian can not be separated into functions of x_i. The Alternating Direction Method of Multipliers (ADMM) addresses this problem and maintains separability while using the augmented Lagrangian to improve convergence through the following updates

$$x_1^{k+1} = \arg\min_{x_1} L_\gamma(x_1, x_2^k, \omega^k) \tag{12.14}$$

$$x_2^{k+1} = \arg\min_{x_2} L_\gamma(x_1^{k+1}, x_2, \omega^k) \tag{12.15}$$

$$\omega^{k+1} = \omega^k + \alpha^k(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b) \tag{12.16}$$

We now consider the quantile regression trend filtering problem,

$$\text{minimize } \rho_\tau(y - \theta) + \lambda||D^{(k)}\theta||_1 \tag{12.17}$$

where $y, \theta \in \mathbf{R}^n$, y is the observed data, $\rho_\tau(u) = \sum_i(\tau - I(u_i < 0))u_i$ is the check loss function corresponding to quantile level τ and $D^{(k)}$ is the discrete differencing matrix of order k and λ is a tuning parameter that controls smoothness. We first re-parameterize $\phi = y - \theta$ so the problem is

$$\text{minimize } \rho_\tau(\phi) + \lambda||D^{(k)}(y - \phi)||_1 \tag{12.18}$$

We further divide ϕ order to solve smaller problems: Defining

$$\phi_1 = (\phi_{11}, \phi_{12}) \tag{12.19}$$

$$\phi_2 = (\phi_{21}, \phi_{22}, \phi_{23}) \tag{12.20}$$

$$\phi_3 = (\phi_{31}, \phi_{32}) \tag{12.21}$$

$$\phi = (\phi_{11}, \phi_{12} = \phi_{21}, \phi_{22}, \phi_{23} = \phi_{31}, \phi_{32}) \tag{12.22}$$

$$\tag{12.23}$$

Dividing y similarly, the problem then becomes

$$\text{minimize } \sum_{i=1}^{3} \rho_\tau(\phi_i) + \lambda||D^{(k)}(y_i - \phi_i)||_1 \tag{12.24}$$

$$\text{subject to: } \phi_{12} = \phi_{21}, \quad \phi_{23} = \phi_{31} \tag{12.25}$$

$$\tag{12.26}$$

We can further simplify by defining

$$\overline{\phi} = (\phi_{11}, \frac{\phi_{12}+\phi_{21}}{2}, \phi_{22}, \frac{\phi_{23}+\phi_{31}}{2}, \phi_{32}) \tag{12.27}$$

$$\overline{\phi_1} = (\phi_{11}, \frac{\phi_{12}+\phi_{21}}{2}) \tag{12.28}$$

$$\overline{\phi_2} = (\frac{\phi_{12}+\phi_{21}}{2}, \phi_{22}, \frac{\phi_{23}+\phi_{31}}{2}) \tag{12.29}$$

$$\overline{\phi_3} = (\frac{\phi_{23}+\phi_{31}}{2}, \phi_{32}) \tag{12.30}$$

so the problem becomes

$$\text{minimize } \sum_{i=1}^{3} \rho_\tau(\phi_i) + \lambda||D^{(k)}(y_i - \phi_i)||_1 \tag{12.31}$$

$$\text{subject to: } \phi_i = \overline{\phi_i} \tag{12.32}$$

$$\tag{12.33}$$

The augmented Lagrangian for this problem is

$$L_\gamma(\phi_1, \phi_2, \phi_3, \overline{\phi_1}, \overline{\phi_2}, \overline{\phi_3}, \omega) = \tag{12.34}$$

$$\sum_{i=1}^{3} \rho_\tau(\phi_i) + \lambda||D^{(k)}(y_i - \phi_i)||_1 + \omega_i^T(\phi_i - \overline{\phi_i}) + \frac{\gamma}{2}||\phi_i - \overline{\phi_i}||_2^2 \tag{12.35}$$

The ADMM updates are then given by

$$\phi_i^{k+1} = \arg\min_{\phi_i} \rho_\tau(\phi_i) + \lambda||D^{(k)}(y_i - \phi_i)||_1 + \omega_i^{kT}(\phi_i - \overline{\phi_i}^k) + \frac{\gamma}{2}||\phi_i - \overline{\phi_i}^k||_2^2 \tag{12.36}$$

$$\omega_i^{k+1} = \omega_i^k + \gamma(\phi_i^{k+1} - \overline{\phi_i}^{k+1}) \tag{12.37}$$

The ϕ_i updates can be obtained using a quadratic program solver such as Gurobi and can be obtained in parallel.

Figure 3: Windows fit separately compared to simultaneous fit, no signal present.

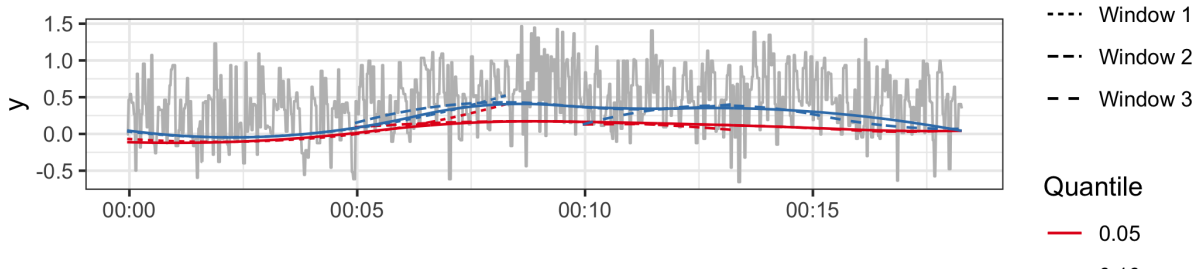


Figure 4: Windows fit with 150 iterations of ADMM, no signal present.

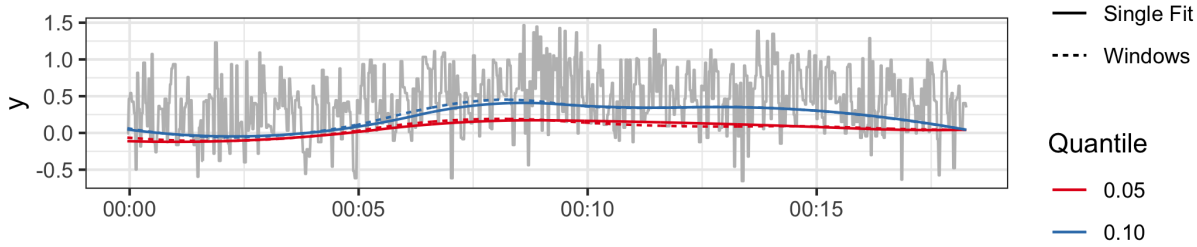


Figure 5: Windows fit separately compared to simultaneous fit, signal present.

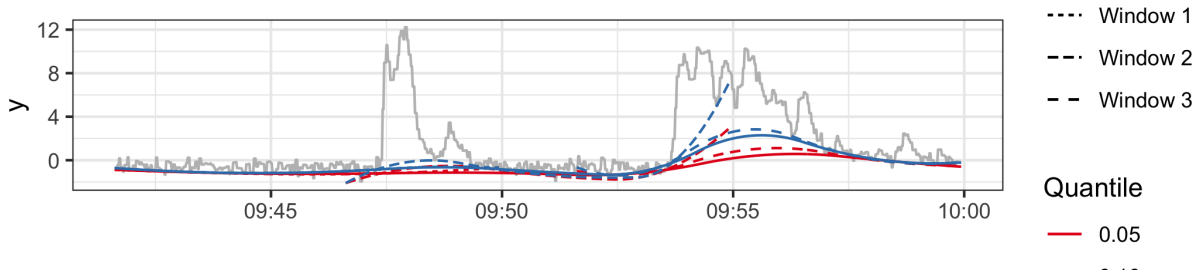
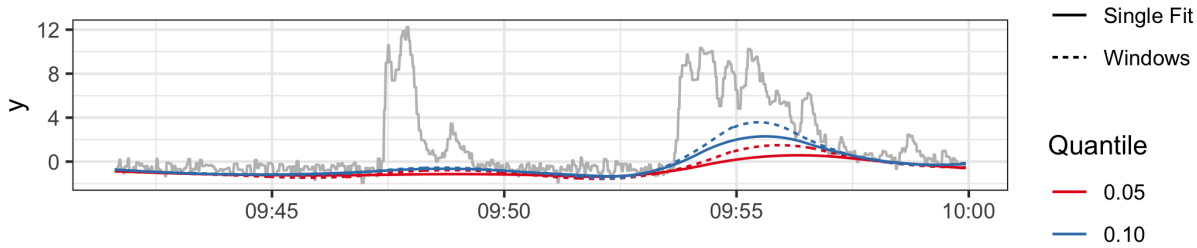


Figure 6: Windows fit with 300 iterations of ADMM, signal present.



13 Cross Validation Methods

Our method can easily handle missing data by changing the check loss function to output 0 for missing values. This allows us to leave out validation observations that can be used to select the tuning parameter λ and to compare method performance on real data. A number of methods have been proposed for selecting the quantile regression smoothing spline tuning parameter Yuan (2006). The traditional smoothing spline for $y \in \mathbf{R}^n$ as a function of a predictor $x \in \mathbf{R}^n$ using the squared loss is estimated by minimizing

$$\frac{1}{n}\sum_i (y_i - f(x_i))^2 + \lambda \int (f''(x))^2 dx \tag{13.1}$$

It has been shown that the solution to this problem is a cubic spline with a knot at each of the observed x_i. This problem can then be reformulated using a spline basis matrix $B \in \mathbf{R}^{nxn}$:

$$(y - B\beta)^T (y - B\beta) + n\lambda \beta^T \Omega \beta \tag{13.2}$$

where $\Omega_{jk} = \int B_j''(x) B_k''(x) dx$. Since each spline basis function is a defined as a cubic polynomial with non-zero second derivatives on a finite region, the integrals are finite. The solution is then given by

$$f(x) = B\widehat{\beta} = B(B^T B + n\lambda\Omega)^{-1} B^T y$$

Because the solution is a linear combination of the observed y, the leave-one-out cross validation (LOOCV) statistics can be calculated without refitting the model. Defining $W = B\widehat{\beta} = B(B^T B + n\lambda\Omega)^{-1} B$ the LOOCV residual is given by

$$y_i - \widehat{y}_i^{(-i)} = \frac{y_i - \widehat{y}_i}{1 - W_{[i,i]}} \tag{13.3}$$

where $\widehat{y}_i$ is the prediction using the full dataset and $\widehat{y}_i^{(-i)}$ is the prediction from the model fit without y_i. In practice, generalized cross validation (GCV) has also been found to be an effective metric for choosing the smoothing parameter, with the validation MSE calculated as

$$\frac{\frac{1}{n}\sum (y_i - \widehat{y}_i)^2}{(1 - \frac{1}{n} tr(W))^2} \tag{13.4}$$

Returning to the quantile regression case, Nychka et al. (1995) proposed a pseudo-data algorithm for a quantile spline estimator of the form

$$\sum_i \rho_\tau(y_i - g(x_i)) + \lambda \int (g''(x))^2 dx. \tag{13.5}$$

If $\rho_\tau(\cdot)$ were differentiable, the solution to this equation would take a form similar to that of the squared loss smoothing spline with weights equal to $\frac{\rho'_\tau(y_i - g(x_i))}{2(y_i - g(x_i))}$. Relying on this idea, Nychka proposed to solve the problem by iteratively solving the weighted smoothing spline. To address the non-differentiability they propose an approximation

$$\rho_{\tau,\delta}(u) = [\tau I(u > 0) + (1 - \alpha) I(u < 0)] u^2/\delta \tag{13.6}$$

They propose a corresponding approximate cross validation metric for choosing the tuning parameter:

$$ACV(\lambda) = \frac{1}{n} \sum_i \rho_\tau \left(\frac{y_i - \widehat{g}_\lambda(x_i)}{1 - h_{ii}} \right) \tag{13.7}$$

where $h_{ii} = \frac{\partial \widehat{g}_{\tau,\lambda}(x_i)}{\partial y_i}$. However Yuan (2006) showed through simulation that this is not a good approximation to the robust cross validation score

$$RCV(\lambda) = \frac{1}{n} \sum \rho_\tau(y_i - g_\lambda^{(-i)}(x_i)) \tag{13.8}$$

and propose the GACV as an alternative

$$GACV(\lambda) = \sum_i \rho_\tau \left(\frac{y_i - \widehat{g}_\lambda(x_i)}{n - tr(H)} \right) \tag{13.9}$$

where $H_{ij} = \frac{\partial \widehat{g}_{\tau,\lambda}(x_i)}{\partial y_j}$

Koenker et al. (1994) proposed a quantile smoothing spline using the check loss function and 1-norm rather than 2-norm:

$$R_{\tau,\lambda} = \rho_\tau(y - g(x)) + \lambda \int_0^1 |g''(x)| dx \tag{13.10}$$

with $0 = x_0 < ... < x_n = 1$. They prove that the resulting function $\widehat{g(x)}$ is a linear spline with knots at the observed values of x. They also argue that the solutions $\widehat{g}_{\tau,\lambda}(\cdot)$ are piecewise constant in λ and that there exists a mesh $0 = \lambda_0 < \lambda_1 < ... < \lambda_J$ such that $\widehat{g}_{\tau,\lambda}(\cdot)$ is the solution for all $\lambda \in [\lambda_{i-1}, \lambda_i]$. Relating λ to the number of interpolated points $p_\lambda = \sum I(y_i = \widehat{g}_i(x_i))$, which can be thought of as active knots, they propose the Schwarz criterion for the selection of λ

$$SIC(p_\lambda) = log[n^{-1} \sum \rho_\tau(y_i - \widehat{g}(x_i))] + \frac{1}{2n} p_\lambda log n \tag{13.11}$$

14 Simulation Study

Methods

We compare the performance of our quantile trend filtering method with the three previously published methods using designs proposed by Racine and Li (2017). The methods compared are

- `npqw`: Racine and Li (2017) constrain the response to follow a smooth location scale model of the form $Y_i = a(X_i) + b(X_i)\epsilon_i$. They estimate the τ_{th} conditional quantile given $X_i = x$ using a kernel estimator

$$q_\tau(x) = \frac{\sum_{i=1}^{n} \Phi^{-1}_{(Y_i, b(X_i))}(\delta_0) K_h(X_i, x)}{\sum_{i=1}^{n} K_h(X_i, x)} \tag{14.1}$$

  defining $\Phi^{-1}_{(Y_i, b(X_i))}(\delta_0)$ as the quantile function of the Normal distribution with mean Y_i and standard deviation $b(X_i)$ evaluated at τ. δ_0 is a function of τ and chosen empirically, h is a tuning parameter and K is a kernel function. Code was obtained from the author for the `quantile-ll` method.

- `qsreg`: Oh et al. (2011) proposed a pseudo-data algorithm for a quantile spline estimator of the form

$$\sum_i \rho_\tau(y_i - g(x_i)) + \lambda \int (g''(x))^2 dx. \tag{14.2}$$

  If $\rho_\tau(\cdot)$ were differentiable, the solution to this equation would take a form similar to that of the squared loss smoothing spline with weights equal to $\frac{\rho'_\tau(y_i - g(x_i))}{2(y_i - g(x_i))}$. Relying on this idea, Nychka proposed to solve the problem by iteratively solving the weighted smoothing spline. To address the non-differentiability they propose an approximation

$$\rho_{\tau,\delta}(u) = [\tau I(u > 0) + (1 - \alpha) I(u < 0)] u^2 / \delta \tag{14.3}$$

  The function `qsreg` in the `fields` R package was used. The smoothing parameter is chosen automatically using generalized cross validation on the pseudo data.

- `rqss`: Koenker et al. (1994) Koenker proposed smoothing splines using trend filtering with the second order differencing matrix which results in linear splines. The function

`rqss` in the `quantreg` package implements this method. The smoothing parameter λ is chosen using a grid search and minimizing

$$SIC(p_\lambda) = log[n^{-1} \sum \rho_\tau(y_i - \widehat{g}(x_i))] + \frac{1}{2n} p_\lambda log n \tag{14.4}$$

where $p_\lambda = \sum I(y_i = \widehat{g}_i(x_i))$, which can be thought of as active knots.

- `detrendr_SIC`: Our method where we minimize $\sum_i \rho_\tau(y_i - \theta_i) + \lambda ||D\theta||_1$ and λ is chosen using SIC from above. A single value of λ was chosen by scaling and summing SIC values across all quantiles.

- `detrendr_valid`: Our method where lambda is chosen by leaving out every 5th observation as a validation data set and evaluating the check loss function on the validation data.

- `detrendr_eBIC`: The traditional BIC is given by

$$\text{BIC}(s) = -2 \log(L\{\hat{\theta}(s)\}) + \nu(s) \log n \tag{14.5}$$

where $\theta(s)$ is the parameter θ with those components outside s being set to 0, and $\nu(s)$ is the number of components in s. If we assume an asymmetric Laplace likelihood $L(y|\theta) = \left(\frac{\tau^n(1-\tau)}{\sigma}\right)^n \exp\left\{-\sum_i \rho_\tau(\frac{y_i - \theta_i}{\sigma})\right\}$ and the number of non-zero elements of $D\theta$ as df

$$\text{BIC}(df) = 2 \sum_i \frac{1}{\sigma} \rho_\tau(y_i - \theta_i) + df \log n \tag{14.6}$$

We can choose and $\sigma > 0$ and have found empirically that $\sigma = \frac{1-|1-2\tau|}{2}$ produces stable estimates. Chen and Chen (2008) proposed the extended BIC for large parameter spaces

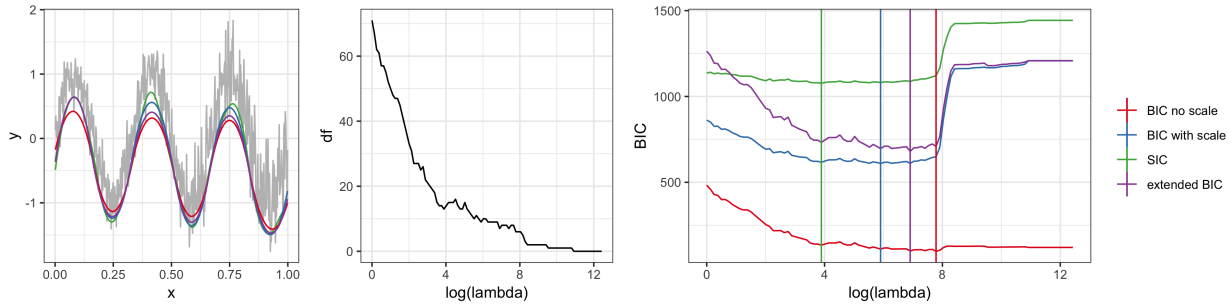
$$BIC_\gamma(s) = -2 \log(L\{\hat{\theta}(s)\}) + \nu(s) \log n + 2\gamma \log \binom{P}{j,} \quad \gamma \in [0, 1] \tag{14.7}$$

where P is the total number of possible parameters and j is the number of parameters included in given model. We used this criteria with $\gamma = 1$, $P = n - k$ where k is the order of the differencing matrix and $j = \nu(s)$ is the number of non-zero entries in $D^{(k)}\theta$.

BIC examples

We used a single dataset to illustrate the difference between the scaled, unscaled and extended BIC criteria.

Figure 7: Degrees of freedom (number of non-zero elements of $D\theta$) by $\log(\lambda)$.



Design

Three simulation designs from Racine and Li (2017) were considered. For all designs X_i was generated as a uniformly spaced sequence in $[0, 1]$ and the response Y was generated as

$$Y_i = sin(2\pi x_i) + \epsilon_i(x_i)$$

The three error distributions considered were

- Gaussian: $\epsilon_i(x_i) \sim N\left(0, \left(\frac{1+x_i^2}{4}\right)^2\right)$
- Beta: $\epsilon_i \sim Beta(1, 11 - 10x_i)$
- Mixed normal: ϵ_i is simulated from a mixture of $N(-1, 1)$ and $N(1, 1)$ with mixing probability x_i.

100 datasets were generated of sizes 300, 500 and 1000. The MSE was calculated as $\frac{1}{n}\sum_i(\hat{q}_\tau(x_i) - q_\tau(x_i))^2$. The plots below show the mean MSE $\pm$ twice the standard error by method, quantile level and sample size.

Figure 8: Simulated data with true quantiles $\tau \in \{0.01, 0.05, 0.25, 0.5, .75, 0.95, 0.99\}$

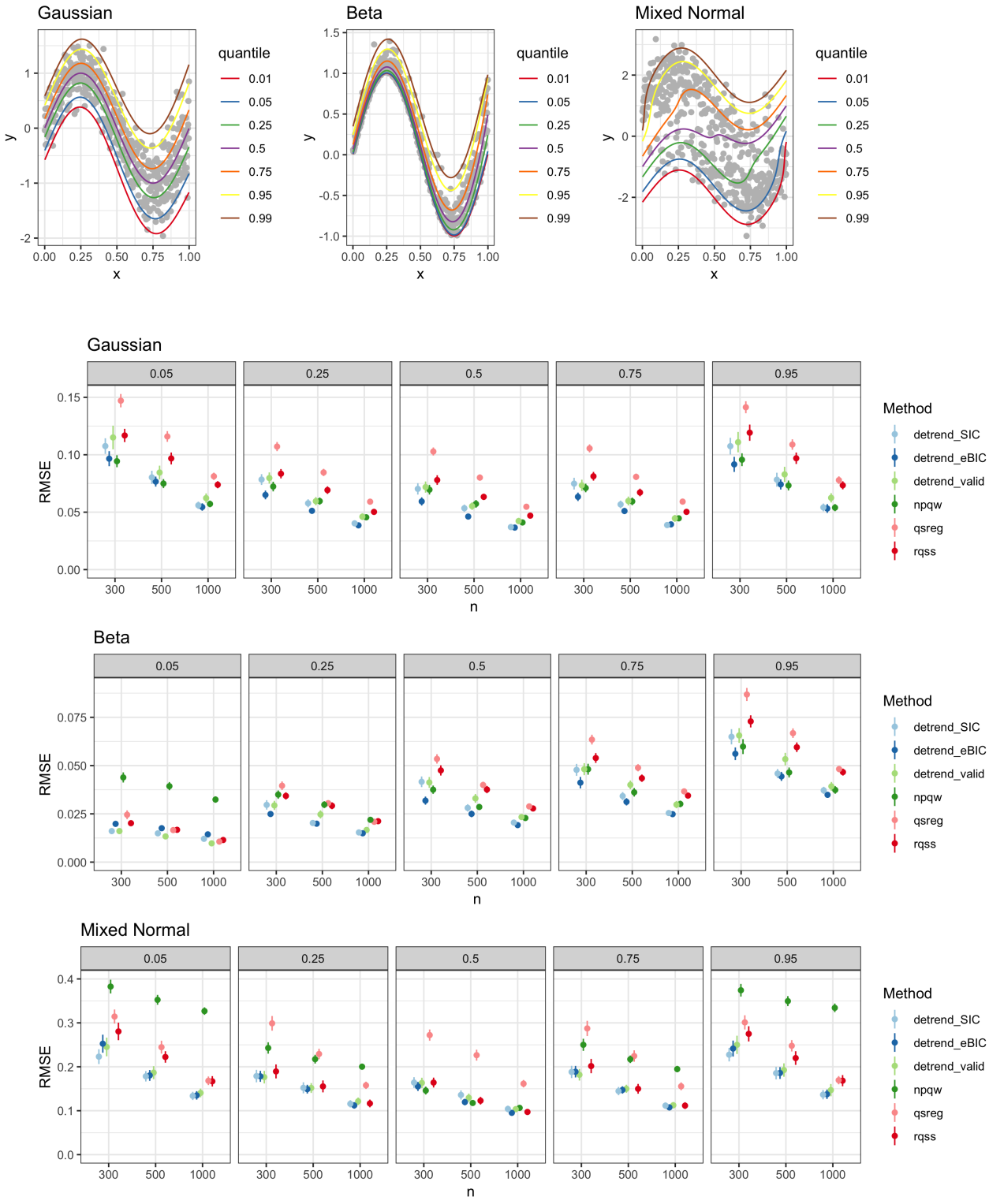
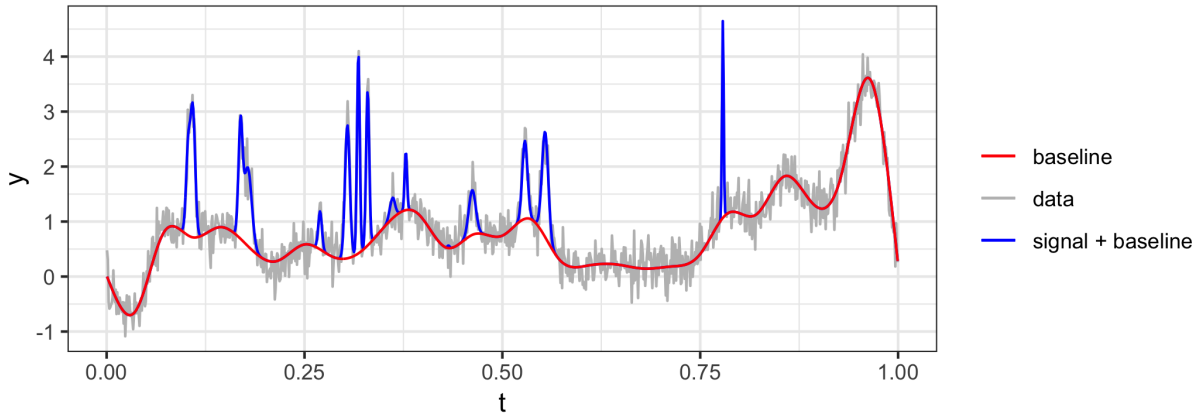


Figure 9: Example of simulated peaks, baseline, and observed measurements.



15 Peaks Simulation

We use another simulation design based on the applied problem we aim to solve. We assume that the measured data can be represented by

$$Y(t) = s(t) + b(t) + \epsilon \tag{15.1}$$

where $s(t)$ is the true signal at time t, $b(t)$ is the drift component that varies smoothly over time and $\epsilon \sim N(0, \sigma^2)$ is an error component. We assume t is a uniformly spaced sequence between 0 and 1. We generate $b(t)$ using a cubic natural spline basis function with degrees of freedom sampled from $n/50$ to $n/25$ with equal probability, and coefficients drawn from an exponential distribution with rate 1. The true signal function is assumed to be zero with Gaussian peaks. The number of peaks is sampled from $n/100$ to $n/50$ with equal probability with centers uniformly distributed between 0.1 and 0.9 and bandwidths uniformly distributed between $1/n$ and $5/n$ and areas uniformly distributed between 0 and $20/n$. One hundred datasets were generated for $n = \{300, 500, 1000, 5000\}$. We compare the methods ability to estimate the true quantiles of $b(t) + \epsilon$ for $\tau \in \{0.01, 0.05, 0.1\}$ and calculate the RMSE.

16 Application

Figure 10: RMSEs compared to the simulated baseline function.

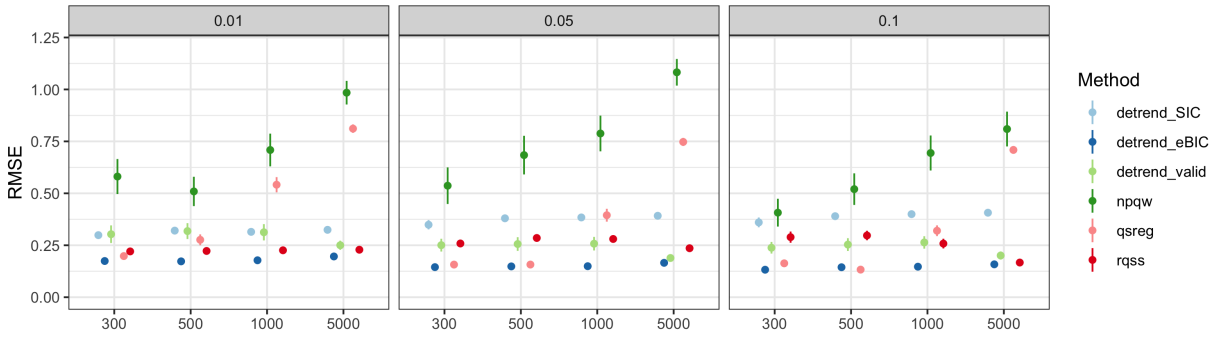


Figure 11: Example signal classification using threshold. Red indicates true signal ¿ 0.1, blue indicates classified as signal after baseline removal using eBIC detrendr and a threshold of 1.

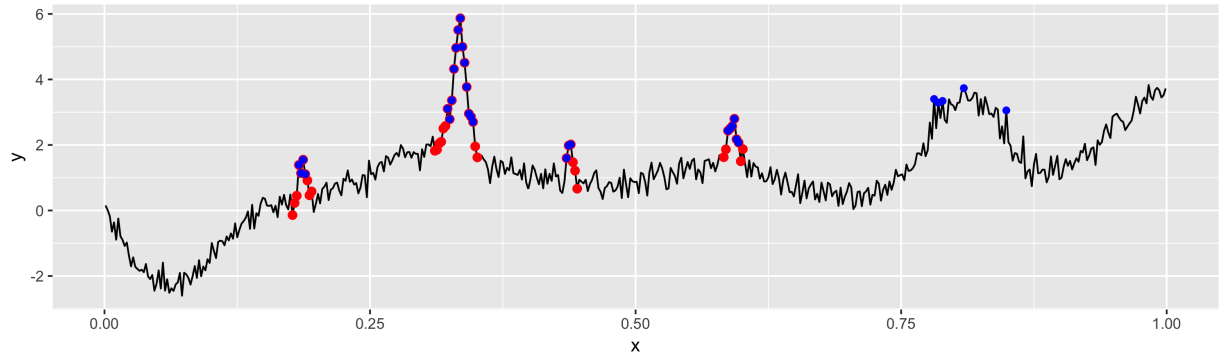


Figure 12: Miss-classification rates by threshold and method, values above the upper limit (npqw) not shown.

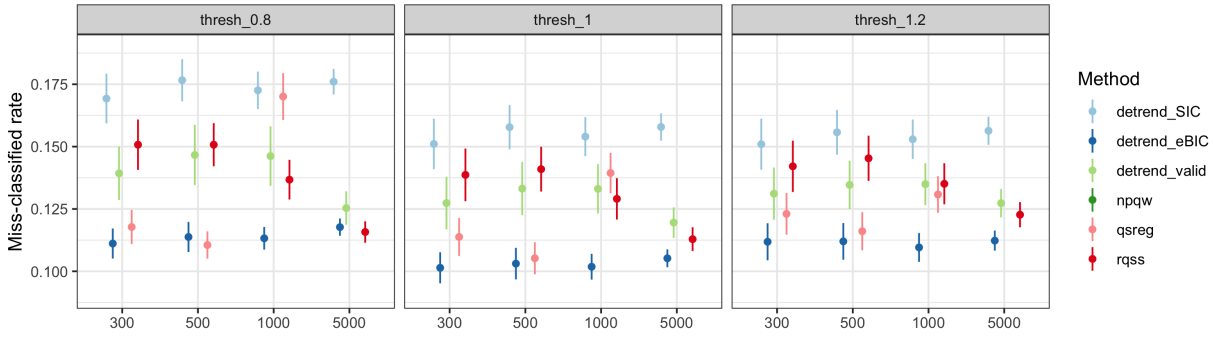
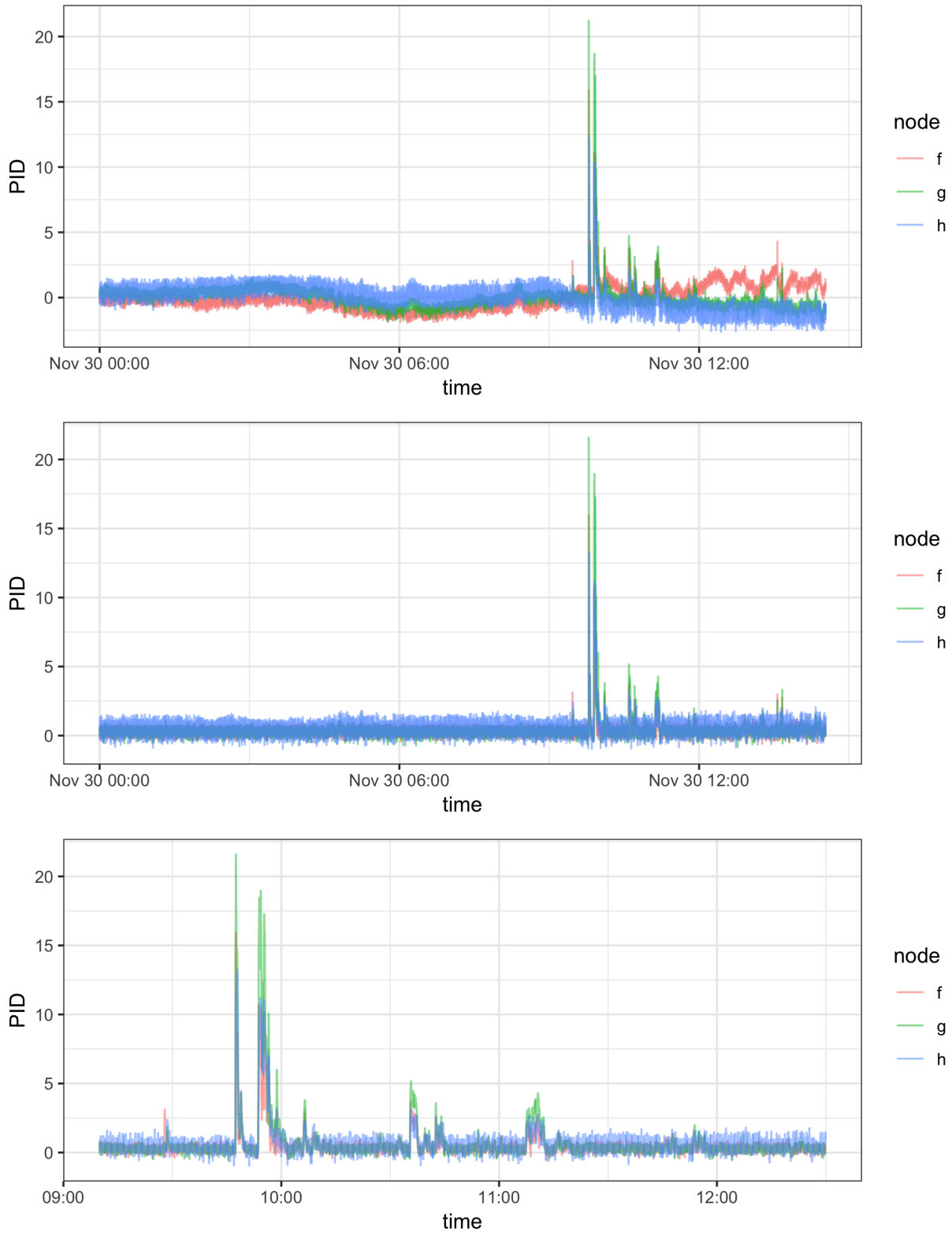


Figure 13: Example of simulated peaks, baseline, and observed measurements.



References

Apte, J. S., Messier, K. P., Gani, S., Brauer, M., Kirchstetter, T. W., Lunden, M. M., Marshall, J. D., Portier, C. J., Vermeulen, R. C., and Hamburg, S. P. (2017), "High-Resolution Air Pollution Mapping with Google Street View Cars: Exploiting Big Data," *Environmental Science & Technology*, in press.

Bacher, R., Chatelain, F., and Michel, O. (2016), "An adaptive robust regression method: Application to galaxy spectrum baseline estimation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4423–4427.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011), "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, 3, 1–122.

Chen, J. and Chen, Z. (2008), "Extended Bayesian information criteria for model selection with large model spaces," *Biometrika*, 95, 759–771.

Combettes, P. L. and Wajs, V. R. (2005), "Signal Recovery by Proximal Forward-Backward Splitting," *Multiscale Modeling & Simulation*, 4, 1168–1200.

Condat, L. (2013), "A Direct Algorithm for 1-D Total Variation Denoising," *IEEE Signal Processing Letters*, 20, 1054–1057.

Davies, P. L. and Kovac, A. (2001), "Local Extremes, Runs, Strings and Multiresolution," *Annals of Statistics*, 29, 1–65.

Davis, D. (2017), "Convergence rate analysis of the forward-Douglas-Rachford splitting scheme," *SIAM Journal on Optimization*, in press.

Eddelbuettel, D. and Sanderson, C. (2014), "RcppArmadillo: Accelerating R with high-performance C++ linear algebra," *Computational Statistics and Data Analysis*, 71, 1054–1063.

He, B. and Yuan, X. (2012), "On the $O(1/n)$ Convergence Rate of the Douglas-Rachford Alternating Direction Method," *SIAM Journal on Numerical Analysis*, 50, 700–709.

— (2015), "On the convergence rate of Douglas-Rachford operator splitting method," *Mathematical Programming*, 153, 715–722.

Ilewicz, W., Kowalczuk, M., Niezabitowski, M., Buchczik, D., and Głuszka, A. (2015), "Comparison of baseline estimation algorithms for chromatographic signals," in *2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 925–930.

Johnson, N. A. (2013), "A Dynamic Programming Algorithm for the Fused Lasso and L0-Segmentation," *Journal of Computational and Graphical Statistics*, 22, 246–260.

Kim, S.-J., Koh, K., Boyd, S., and Gorinevsky, D. (2009), "ℓ_1 Trend Filtering," *SIAM Review*, 51, 339–360.

Koenker, R. and Bassett, G. (1978), "Regression Quantiles," *Econometrica*, 46, 33–50.

Koenker, R., Ng, P., and Portnoy, S. (1994), "Quantile smoothing splines," *Biometrika*, 81, 673–680.

Ning, X., Selesnick, I. W., and Duval, L. (2014), "Chromatogram baseline estimation and denoising using sparsity (BEADS)," *Chemometrics and Intelligent Laboratory Systems*, 139, 156 – 167.

Nychka, D., Gray, G., Haaland, P., Martin, D., and O'connell, M. (1995), "A nonparametric regression approach to syringe grading for quality improvement," *Journal of the American Statistical Association*, 90, 1171–1178.

Oh, H.-S., Lee, T. C. M., and Nychka, D. W. (2011), "Fast Nonparametric Quantile Regression With Arbitrary Smoothing Methods," *Journal of Computational and Graphical Statistics*, 20, 510–526.

Racine, J. S. and Li, K. (2017), "Nonparametric conditional quantile estimation: A locally weighted quantile kernel approach," *Journal of Econometrics*, 201, 72–94.

Ramdas, A. and Tibshirani, R. J. (2016), "Fast and Flexible ADMM Algorithms for Trend Filtering," *Journal of Computational and Graphical Statistics*, 0, 0–0.

Sanyal, A., Baral, A., and Lahiri, A. (2012), "Application of S-transform for removing baseline drift from ECG," in *2012 2nd National Conference on Computational Intelligence and Signal Processing (CISP)*, pp. 153–157.

Spingarn, J. E. (1985), "Applications of the method of partial inverses to convex programming: Decomposition," *Mathematical Programming*, 32, 199–223.

Takeuchi, I., Le, Q. V., Sears, T. D., and Smola, A. J. (2006), "Nonparametric quantile estimation," *Journal of Machine Learning Research*, 7, 1231–1264.

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

Walker, H. F. and Ni, P. (2011), "Anderson Acceleration for Fixed-Point Iterations," *SIAM Journal on Numerical Analysis*, 49, 1715–1735.

Yuan, M. (2006), "GACV for quantile smoothing splines," *Computational statistics & data analysis*, 50, 813–829.