Repeated median and hybrid filters

Roland Fried[a], Thorsten Bernholt[b] and Ursula Gather[a]

[a] *Department of Statistics, University of Dortmund, 44221 Dortmund, Germany*
[b] *Department of Computer Science, University of Dortmund, 44221 Dortmund, Germany*

Abstract

Standard median filters preserve abrupt shifts (edges) and remove impulsive noise (outliers) from a constant signal but they deteriorate in trend periods. FIR median hybrid (FMH) filters are more flexible and also preserve shifts, but they are much more vulnerable to outliers. Application of robust regression methods, in particular of the repeated median, has been suggested for removing subsequent outliers from a signal with trends. A fast algorithm for updating the repeated median in linear time using quadratic space is given in Bernholt and Fried (2003). We construct repeated median hybrid filters to combine the robustness properties of the repeated median with the edge preservation ability of FMH filters. An algorithm for updating the repeated median is presented which needs only linear space. We also investigate analytical properties of these filters and compare their performance via simulations.

Keywords: Signal extraction; Drifts; Jumps; Outliers; Update algorithm

1 Introduction

Robust filtering of online monitoring data is a challenging task. In intensive care the heart rate and blood pressure are measured in short time lags which provides important information for bedside decision support. The extraction of the underlying, clinically relevant signal from the observed time series is a basic task. A solution needs to preserve abrupt level shifts and monotonic trends, while irrelevant short-term fluctuations and outliers caused by e.g. patient's movements or measurement artifacts should be removed.

The extraction of a signal (μ_t) from a time series (x_t) can be formalized using a components model like

$$x_t = \mu_t + u_t + v_t,\ t \in \mathbb{Z}. \tag{1}$$

Here, μ_t is the smoothly varying level of the time series, possibly with a few abrupt shifts, which we want to extract from the data, u_t is additive noise from a symmetric distribution with mean zero and variance σ^2, and v_t is impulsive (spiky) noise from some outlier generating mechanism.

Linear filters like moving averages track trends and are very efficient for Gaussian noise (u_t), but they are highly vulnerable to impulsive noise (outliers or spikes) and they blur level shifts (also called step changes, edges or jumps). Standard median filters remove outliers and preserve level shifts (Tukey, 1977, Nieminem, Neuvo and Mitra, 1989), but their output does not properly represent linear trends. FIR median hybrid (FMH) filters (Heinonen and Neuvo, 1987, 1988) combine advantages of linear and nonlinear filters by taking the median value of several linear FIR subfilters. The linear subfilters can be designed to track polynomial trends well, and the use of the central observation as central subfilter allows to preserve level shifts similarly good as standard median filters. Davies, Fried and Gather (2004) suggest application of a robust regression estimator like the repeated median to overcome the difficulties of the median in trend periods.

In this paper we investigate robust hybrid filters, which are designed to combine the good properties of FMH filters with the robustness of the repeated median in trend periods. Instead of linear subfilters we use robust counterparts, namely half-window medians for a constant signal and half-window repeated medians for a linear trend. For the central subfilter, we can apply either the central observation or the median value of the whole time window. We investigate analytical properties of the resulting hybrid filters and compare their ability to attenuate Gaussian noise, to resist outliers and to preserve abrupt shifts in case of a constant signal and in a trend period via simulations.

In Section 2 we review repeated median and FMH filters and introduce repeated median hybrid filters as a combination of these. In Section 3 we discuss the fast computation of the filters and propose a new algorithm for updating the repeated median. In Section 4 we investigate the filters from a theoretical point of view. In Section 5 we compare their performance in basic data situations via simulations. In Section 6 we present some applications to real and simulated data for further comparison before we end up drawing some conclusions.

2 Hybrid filters

We introduce some classes of nonlinear filters. Let the filter input be a time series (x_t). Moving averages and standard median filters are based on the idea to approximate the signal (μ_t) underlying (x_t) by moving a time window of fixed width $n = 2k+1$ through the series to approximate the true level of the time series in the center of the window. Hence, applying any such method causes a time delay of k observations. A short delay can be achieved by a small choice of k, but for the cost of reduced smoothing.

2.1 Filters based on robust regression

Standard median filters (running medians) approximate the signal μ_t in the center of a moving time window $\{x_{t-k}, \ldots, x_{t+k}\}$ by the median of these observations,

$$SM(x_t) \;=\; \tilde{\mu}_t = med\{x_{t-k}, \ldots, x_{t+k}\},\; t \in \mathbb{Z}.$$

This means to treat the level μ_t as locally almost constant. This assumption is of course not appropriate in trend periods. Therefore, Davies et al. (2004) suggest to fit a local linear trend $\mu_{t+i} = \mu_t + i\beta_t$, $i = -k, \ldots, k$, by robust regression. Here, μ_t is again the level and β_t is the slope in the center of the time window. Based on a comparison of methods with high breakdown point they recommend Siegel's (1982) repeated median defined by

$$\begin{aligned} RM(x_t) \;=\; \tilde{\mu}_t^{RM} &= med\{x_{t-k} + k\tilde{\beta}_t, \ldots, x_{t+k} - k\tilde{\beta}_t\} \\ \tilde{\beta}_t^{RM} &= med_{i=-k,\ldots,k}\left\{ med_{j=-k,\ldots,k, j\neq i} \frac{x_{t+i} - x_{t+j}}{i-j} \right\}. \end{aligned}$$

The slope within the time window is estimated by taking repeated medians of pairwise slopes, namely an inner median with one observation being fixed, and then the median of all these inner medians. Bernholt and Fried (2003) present an algorithm for updating the repeated median (RM) filter in linear time using quadratic space when moving the time window by one step, which reduces the computational complexity of a straightforward implementation substantially.

2.2 Linear median hybrid filters

Linear median hybrid filters have been suggested to combine the good properties of linear and median filters by linear and nonlinear operations (Heinonen and Neuvo, 1987, 1988, Astola, Heinonen and Neuvo, 1989). They are computationally much less expensive than standard median filters and offer increased flexibility due to the use of linear subfilters Φ_j, $j = 1, \ldots, M$, which are applied to the input data before taking the median of the outcomes. When the subfilters respond to a finite number of impulses only the resulting procedure is called linear median hybrid filter with finite impulse response, more briefly FIR median hybrid filter or FMH filter:

$$FMH(x_t) = med\{\Phi_1(x_t), \Phi_2(x_t), \ldots, \Phi_M(x_t)\}, \quad t \in \mathbb{Z},$$

where $\Phi_1(x_t), \ldots, \Phi_M(x_t)$ are the outputs of the FIR filters. The proper choice of M and of the subfilters depend on the type of signal and on the demands specified by the operator. FMH filters can even be less biased at edges than the standard median (Astola et al., 1989).

Heinonen et al. (1985) and Kalli et al. (1985a, 1985b) filter blood pressure measurements by a simple FMH filter with $M = 3$ subfilters, namely two one-sided moving-averages and the current observation x_t as central subfilter for edge preservation:

$$\Phi_1(x_t) = \frac{1}{k}\sum_{i=1}^{k} x_{t-i}, \quad \Phi_2(x_t) = x_t, \quad \Phi_3(x_t) = \frac{1}{k}\sum_{i=1}^{k} x_{t+i}.$$

Like the standard median this FMH filter implicitly takes the signal as locally almost constant since ordinary location estimates are applied. Instead of moving averages, predictive FMH filters apply FIR subfilters for one-sided extrapolation of a trend:

$$\begin{aligned} PFMH(x_t) &= med\{\Phi_F(x_t), x_t, \Phi_B(x_t)\}, \\ \Phi_F(x_t) = \sum_{i=1}^{k} h_i x_{t-i}, &\qquad \Phi_B(x_t) = \sum_{i=1}^{k} h_i x_{t+i}. \end{aligned}$$

Heinonen and Neuvo (1988) choose the weights h_i to obtain optimal (in the sense of mean square error MSE) predictions for a polynomial signal which is disturbed by Gaussian white noise imposing the restriction that the exact signal value is obtained in the deterministic case without noise. For a polynomial of degree zero corresponding to a constant signal this reduces to the simple FMH filter given above. For a polynomial of degree one, i.e. a linear trend $\mu_{t+i} = \mu_t + i\beta_t$, the resulting weights are $h_i = \frac{4k-6i+2}{k(k-1)}$, $i = 1, \ldots, k$. Another possibility is to use predictions of different degrees. The combined FMH filter

$$CFMH(x_t) = med\{\Phi_F(x_t), \Phi_1(x_t), x_t, \Phi_3(x_t), \Phi_B(x_t)\},$$

applies the subfilters $\Phi_1(x_t)$, $\Phi_3(x_t)$, $\Phi_F(x_t)$ and $\Phi_B(x_t)$ for forward and backward extrapolation of a constant signal or a linear trend given above.

2.3 Repeated median hybrid filters

In view of increased computational power computation time is nowadays a much less severe constraint. In order to increase the robustness of the procedure, we can construct hybrid filters with robust instead of linear subfilters.

It is near at hand to replace the one-sided moving averages $\Phi_1(x_t)$ and $\Phi_3(x_t)$ by half-window medians $\tilde{\mu}_t^F = med\{x_{t-k}, \ldots, x_{t-1}\}$ and $\tilde{\mu}_t^B = med\{x_{t+1}, \ldots, x_{t+k}\}$, and the one-sided linear extrapolators $\Phi_F(x_t)$ and $\Phi_B(x_t)$ by robust linear regression estimators $RM^F(x_t)$ and $RM^B(x_t)$, with $RM^F(x_t)$ and $RM^B(x_t)$ denoting the estimate of the level at time t obtained by applying the repeated median to $x_{t-k}, \ldots, x_{t-1}$ and to $x_{t+1}, \ldots, x_{t+k}$, respectively:

$$\begin{aligned}
RM^F(x_t) &= med\{x_{t-k} + k\tilde{\beta}_t^F, \ldots, x_{t-1} + \tilde{\beta}_t^F\} \\
\tilde{\beta}_t^F &= med_{i=-k,\ldots,-1}\left\{med_{j=-k,\ldots,-1, j\neq i}\frac{x_{t+i} - x_{t+j}}{i-j}\right\} \\
RM^B(x_t) &= med\{x_{t+1} - \tilde{\beta}_t^B, \ldots, x_{t+k} - k\tilde{\beta}_t^B\} \\
\tilde{\beta}_t^B &= med_{i=1,\ldots,k}\left\{med_{j=1,\ldots,k, j\neq i}\frac{x_{t+i} - x_{t+j}}{i-j}\right\}.
\end{aligned}$$

In the resulting repeated median hybrid filters we can use the central observation x_t as central subfilter for edge preservation like in the FMH filters or we can replace x_t by the median $\tilde{\mu}_t$ of the time window. We investigate both possibilities in the following and call them RMH and RMMH filters, respectively. Moreover, we will consider the robust predictive and robust combined variants

$$\begin{aligned}
PRMH(x_t) &= med\{RM^F(x_t), x_t, RM^B(x_t)\} \\
CRMH(x_t) &= med\{RM^F(x_t), \tilde{\mu}_t^F, x_t, \tilde{\mu}_t^B, RM^B(x_t)\} \\
PRMMH(x_t) &= med\{RM^F(x_t), \tilde{\mu}_t, RM^B(x_t)\} \\
CRMMH(x_t) &= med\{RM^F(x_t), \tilde{\mu}_t^F, \tilde{\mu}_t, \tilde{\mu}_t^B, RM^B(x_t)\}.
\end{aligned}$$

3 Computation

We briefly explain the basic ideas of efficient implementation of the previous procedures. Signal extraction using a moving time window allows application of algorithms for updating the results from the previous window in order to save computation time.

3.1 Median

An algorithm for updating (or 'maintaining') the standard median is given in Cormen, Leiserson and Rivest (1990, Section 15.1). This algorithm uses a red-black tree for storing the values in the current time window in a sorted order. Deletion and insertion of a value is supported in $O(\log n)$ worst case time using $O(n)$ space. In each node of the tree, the number of nodes in the left respectively right subtree is maintained. Using this information the median in the updated tree can be found in logarithmic time, by starting at the root of the tree and always branching into the appropriate subtree until a leaf of the tree, the median is reached. This reduces the time needed for calculating the median from scratch using Quickselect, which has an average running time of $O(n)$.

3.2 Repeated Median

An algorithm for updating the repeated median in $O(n)$ time using $O(n^2)$ space is presented by Bernholt and Fried (2003) using a hammock graph for storing the data in the dual space. Here, we present a new algorithm which is based on ideas of Stein and Werman (1992) and which needs $O(n)$ space only. Consider the data $(t+i, x_{t+i})$, $i = -k, \ldots, k$, in the current time window along with the observation times as points in the plane. According to the point-line duality the point $(t+i, x_{t+i})$ can be mapped to the dual line ℓ_i defined by $v = (t+i)u + x_{t+i}$ in (u, v) coordinates. In the dual space, we say that the point (u, v) is located *left of* the line ℓ if there exists a constant $c > 0$ such that the point $(u + c, v)$ is located on ℓ. The terms *right of*, *above* and *below* are defined analogously. The pairwise slope $(x_{t+i} - x_{t+j})/(i - j)$ in the calculation of $\tilde{\beta}_t$ is the u-coordinate of the intersection point (u_{ij}, v_{ij}) of the lines ℓ_i and ℓ_j. The solution with parameters $(\tilde{\mu}_t^{RM}, \tilde{\beta}_t^{RM})$ is mapped to the point $(-\tilde{\beta}_t^{RM}, \tilde{\mu}_t^{RM})$, which is the intersection of two lines. We order the lines ℓ_i according to their occurrence on an imaginary vertical line ν at position $-\tilde{\beta}_t$ and denote this permutation of the lines by π. In each updating step, one former line has to be deleted from π and a new one inserted into π. If the solution has changed, the permutation at the new position has to be computed. This is done by moving the imaginary line ν towards the new solution point as displayed in Figure 1.

Figure 1 about here

At each intersection point visited, while moving, the two lines intersecting in this point are transposed within π. We will refer to the number of visited intersection points as S. The direction of the line movement is determined by counting for each line ℓ_i the number c_i of intersection points on ℓ_i to the left of ν. Let $L = \#\{c_i : c_i > \lfloor (n-1)/2 \rfloor\}$ and $R = \#\{c_i : c_i \leq \lfloor (n-1)/2 \rfloor\}$. A new solution $(-\tilde{\beta}_{t+1}^{RM}, \tilde{\mu}_{t+1}^{RM})$ is attained if $L = R$. In the case that two inner medians are located on the same intersection point they are identical and only $|L - R| \leq 1$ can be reached since the corresponding c_i's are both increased or decreased by one if this intersection point is passed. Otherwise, if $L < R$ we need to move to the right, while we need to move to the left if $L > R$.

Figure 2 about here

In order to maintain the permutation π we store the lines in a dictionary T, e.g., a red-black tree or AVL-tree, illustrated in Figure 2 and described in Cormen, Leiserson and Rivest (1990). As the depth of such trees is bounded by $O(\log n)$, these data structures support insertion, deletion and search in $O(\log n)$ time. Note that the next point to be visited, when moving the vertical line, is always an intersection point of two lines which are neighbors in the current permutation π. In order to find the next intersection point in $O(\log n)$ time w.r.t. the ordering of the u-coordinate, we store the intersection points on both sides of the current imaginary line ν in two priority queues Q_L and Q_R e.g. organized as heaps. When moving ν to the next intersection point two lines are swapped, which creates two new neighborhoods and destroys two old ones. Depending on the location of new (old) intersection points w.r.t. ν they need to be inserted (deleted) in the priority queues, which can be done in $O(\log n)$

time. The next intersection point in the priority queue can also be found in $O(\log n)$ time. Insertion (deletion) of a new (old) line in an updating step involves updating the c_i's. Call the position of the inserted (deleted) line π_i. As the inserted (deleted) line ℓ^* has the largest (smallest) slope, all lines stored in $\pi_1, \ldots, \pi_{i-1}$ ($\pi_{i+1}, \ldots, \pi_n$) intersect ℓ^* on the left hand side. Therefore the c_i's for all lines that are stored left (right) of ℓ^* in the tree have to be increased (decreased) by 1. The problem is that there are $O(n)$ c_i's changing their value and that the correct L has to be computed. In order to achieve a running time of $O(\log n)$, we store the following variables in each node of the search tree T:

- $u(v)$: The update value of the node v. The correct c_i of a leaf can be obtained by summing up over all u found on the path from the root of the tree to that leaf.
- $w_a(v)$: The maximal value of a c_i in the subtree of v smaller than $n/2$.
- $a(v)$: The number of c_i in the subtree of v which are equal to w_a.
- $w_b(v)$: The minimal value of a c_i in the subtree of v greater than $n/2$.
- $b(v)$: The number of c_i in the subtree of v which are equal to w_b.

The new line is inserted as a node v^* into the search tree T. Consider the search path P to the node v^*, the boundary $B = \{v' \mid v' \text{ is a child of } v,\ v \in P \text{ and } v' \notin P\}$ and the left boundary $B^\ell = \{v' \mid v' \text{ is a left child of } v,\ v \in P \text{ and } v' \notin P\}$. During the insertion we increase the values u, w_a and w_b in all nodes $v \in B^\ell$ by one. L is changed by ΔL, which is the number of all nodes $v \in P$ where w_a has exceeded $n/2$: $\Delta L = \sum_{v \in P}\{a(v) \mid w_a(v) > n/2\}$. To update the values of w_a and w_b, we call for every $v \in B^\ell$ the following update routine:

Routine Update(v)
Input:
  A node v. Let v_1 and v_2 be the two child nodes.
begin
  (The case that one of v_1 and v_2 does not exist is not explicitly mentioned)
  if $w_a(v) > n/2$ **then**
    $w_b(v) := w_a(v)$ and $b := a$
    if v is a leaf **then**
      $a := 1$ and $w_a(v) := c_i$, that is stored in this node v.
    else
      Add $u(v)$ to the values of $w_a(v_1)$,$w_b(v_1)$,$w_a(v_2)$ and $w_b(v_2)$.
      $u(v) := 0$
      Call Update(v_1) and Update(v_2)
      $w_a(v) := \max\{w_a(v_1), w_a(v_2)\}$

$$a(v) := \begin{cases} a(v_1) + a(v_2) & \text{if } w_a(v_1) = w_a(v_2) \\ a(v_1) & \text{if } w_a(v_1) > w_a(v_2) \\ a(v_2) & \text{otherwise} \end{cases}$$

    end if
  end if
end

The procedure visits all nodes on the ΔL paths towards leaves where the value of the c_i has exceeded $n/2$, thus the computing time for inserting (deleting) a line is $O(\Delta L \cdot \log n)$. As the algorithm has to make at least $\Delta L/2$ steps until the new optimum is reached and one move of the vertical line ν can be computed in $O(\log n)$, the overall running time is bounded by $S \cdot O(\log n)$. In Section 6.1 we study the possible values of S empirically.

The algorithm described above allows updating the regression slope. For updating the extracted signal value we use the fact that

$$RM(x_t) \quad = \quad med_{i=-k,\dots,k} \left\{ med_{j=-k,\dots,k,j\neq i} \frac{jx_{t+i} - ix_{t+j}}{i-j} \right\} .$$

(Stein and Werman, 1992) and we modify the algorithm in such a way that we use an imaginary horizontal line instead.

3.3 Repeated median hybrid filters

The calculation of the repeated median hybrid filters requires obtaining the median of three or five values and updating the backward extrapolating filter(s) as described above, which are stored and reused for calculating the forward extrapolating filters. Using a FIFO queue, which can be implemented as a linked list, the new element $(RM^B(x_t), \tilde{\beta}_t^B)$ is inserted at the head of the list, and $RM^F(x_t)$ is obtained from the previous $(RM^B(x_{t-k-1}), \tilde{\beta}_{t-k-1}^B)$ at the end of the list as

$$RM^F(x_t) = RM^B(x_{t-k-1}) + (k+1)\tilde{\beta}_{t-k-1}^B$$

before deleting $(RM^B(x_{t-k-1}), \tilde{\beta}_{t-k-1}^B)$. All this can be done in $O(1)$ time. The $\tilde{\mu}_t^F$ is got immediately from such a list of the $\tilde{\mu}_t^B$. Using the former computations saves a constant factor. Nevertheless, the running time is dominated by the computation of $RM^B(x_t)$.

3.4 FMH filters

Besides calculating the median of three or five values in $O(1)$ time, updating the FMH filters needs to maintain weighted sums of the observed values. Replacing a value in an ordinary sum for obtaining an updated arithmetic mean can be done in $O(1)$ time. For $\Phi_B(x_t)$ we use

$$\begin{aligned} \Phi_B(x_t) - \Phi_B(x_{t-1}) &= \sum_{i=1}^{k} h_i x_{t+i} - \sum_{i=1}^{k} h_i x_{t+i-1} \\ &= h_k x_{t+k} - h_1 x_t + \frac{6}{k(k-1)} \sum_{i=1}^{k-1} x_{t+i} . \end{aligned}$$

This allows updating the difference which is added to the previous $\Phi_B(x_{t-1})$ in $O(1)$ time.

4 Analytical properties

For a theoretical analysis of the filtering procedures we use the components model (1) and study the behavior in a single time window centered at zero considering the data points $x_{-k}, \dots, x_0, \dots, x_k$ with the same k for all filters.

4.1 Equivariance and invariance

Equivariance and invariance are important properties of statistical procedures. Location equivariance means that adding a constant to all observations (in the time window) changes the resulting estimate accordingly. Similarly, scale equivariance means that multiplication of all observations with a constant changes the extracted signal in the same way. All the previous procedures have both properties.

For tracking linear trends sometimes linear trend preservation is considered, which means that the output of the filter is identical to the measured time series if the observations lie on a straight line. This property is weak and is inherent to all the previous methods. However, it does not consider observational noise, which can hamper the tracking of linear trends. Another property is preferable instead, which we might call trend invariance: The extracted signal should not depend on a trend as long as the central level remains fixed. This means, when varying the trend in the time window replacing $x_{-k}, \ldots, x_k$ by $x_{-k} - k\beta, \ldots, x_{-1} - \beta, x_0, x_1 + \beta, \ldots, x_k + k\beta$, the estimated central level should remain the same.

The RM, the predictive FMH and the predictive RMH are the only trend invariant procedures among those presented in Section 2. The trend invariance of the RM follows directly from the regression equivariance of the underlying regression estimator. When adding a constant trend to some data, the backward and forward subfilters of the predictive FMH and RMH still result in the same extrapolations, and hence the median will be taken from the same values as before.

The standard median is not trend invariant. Consider e.g. the case $k = 1$ and the observations $1, 0, -3$ with median zero. Adding a trend with slope $\beta = 2$ we obtain the sample $-1, 0, -1$ with median -1, i.e. the outcome depends on the underlying slope. In the same way, FMH and RMH filters are not trend invariant when subfilters are used which are not trend invariant like medians or one-sided moving averages.

4.2 Noise-free situation - a best case analysis

The tracking of trends, the preservation of abrupt shifts and the removal of impulsive noise (outliers) are essential properties of robust filters. The best possible performance of a procedure can be inspected in a noise-free situation, where $\sigma^2 = 0$. The standard median preserves a level shift exactly in this idealized situation, and it removes up to k subsequent spikes completely when using the window width $n = 2k + 1$ and the signal is constant otherwise. It even allows to recover monotonic trends, but it preserves a shift in a trend period only if the shift has the same direction as the trend. Otherwise the shift gets blurred, and spikes within a trend cause further undesirable effects.

As opposed to standard medians, FMH filters do not delay a level shift when there is an impulse in the opposite direction to the step within k time points after the shift. However, they suffer from smearing then as the height of a shift and a constant signal value close to the shift change (Heinonen and Neuvo, 1987). Like all methods considered here, FMH filters recover a linear trend exactly in the noise-free situation. The combined FMH removes a single spike and preserves a shift exactly only in case of a constant signal or if the outlier (shift) has the same direction as the trend, while e.g. a negative spike (shift) in an upward trend produces smearing. The predictive FMH removes a single spike and preserves a shift exactly within a linear trend period irrespective of the directions as it is trend invariant.

The RM can remove $k-1$ spikes completely within a single time window as then the median slope for each of the $k+2$ clean data points is calculated from $k+1$ pairs of clean observations. This number is slightly smaller than for the median in case of a constant signal, but for the RM it does not depend on a trend at all.

The repeated median hybrid filters improve on the FMH filters w.r.t. the removal of spikes as they can resist up to $\lfloor k/2 \rfloor$ subsequent outliers without any effect. For the predictive RMMH this number is even $2\lfloor k/2 \rfloor - 1$. In order to see this, we note that for the predictive filters two subfilters giving the exact signal value are sufficient to guarantee that this is also true for the hybrid filter, while for the combined filters it is sufficient that there is one more filter with a smaller and one with a larger outcome. Furthermore, each of the extrapolating RM filters gives the exact signal value if it is influenced by at most $\lfloor k/2 \rfloor - 1$ spikes. Now, if there are only $\lfloor k/2 \rfloor$ spikes at the right hand side of the center, the forward extrapolating subfilter and the central subfilter and hence the RMH and the RMMH give the exact signal value (as the outcome of one of the half-window medians in the combined filters will be at most and the other one at least the signal value). If not all of the $\lfloor k/2 \rfloor$ spikes are at the same side of the center both extrapolating RM filters give the exact signal value. However, more than $\lfloor k/2 \rfloor$ negative outliers at the right of the center in an upward trend can change the backward extrapolating filters and hence the combined RMH and RMMH to become smaller than the signal value, while outliers at positions $0, \ldots, \lfloor k/2 \rfloor$ change the central subfilter and the backward extrapolating RM filter in the predictive RMH arbitrarily. The predictive RMMH, however, gives the exact signal value as long as there are at most $\lfloor k/2 \rfloor - 1$ outliers at one side of the center and k spikes altogether because then one of the extrapolating filters and the median give the correct value. This is guaranteed if there are at most $2\lfloor k/2 \rfloor - 1$ subsequent spikes. Note that the performance of the combined RMMH depends on the sign of the trend and the spikes: The half-window medians are still to the same side of the signal value as in the case without spikes when being affected by $\lfloor k/2 \rfloor$ spikes in the same direction as the trend, e.g. if both are positive. Therefore, the combined RMMH can remove more spikes completely, namely $\lfloor k/2 \rfloor + 1$, if they are in the same direction as the trend. Furthermore, the predictive RMH and RMMH preserve shifts exactly, while for the combined RMH and RMMH this is true only if the shift is in the same direction as the trend, just like for the combined FMH. Table 1 summarizes these results.

Table 1: Number of subsequent spikes removed from a linear trend without noise.

SM	RM	PFMH	CFMH	PRMH	CRMH	PRMMH	CRMMH
0	$k-1$	1	0	$\lfloor k/2 \rfloor$	$\lfloor k/2 \rfloor$	$2\lfloor k/2 \rfloor - 1$	$\lfloor k/2 \rfloor$

4.3 Continuity - small deviations

The results in the previous subsection are for the case without observational noise. Davies (1993) suggests application of continuous, particularly of Lipschitz continuous procedures for restricting the influence of small changes in the data like rounding or a small amount of observational noise.

All the filters investigated here are Lipschitz continuous. The median, like any other order statistic, is Lipschitz continuous with constant 1 as changing every observation by less than a value δ changes the median at most by δ. As every linear filter is Lipschitz continuous with constant $max|h_i|$, the maximal absolute weight, an FMH filter is Lipschitz continuous with constant $max|h_i^j|$, the maximal absolute weight given by any subfilter. The repeated median is Lipschitz continuous with constant $2k+1$: The slope estimate changes at most by 2δ when changing each observation by at most δ, and hence the level estimate is bounded not to change more than $(2k+1)\delta$ as none of the trend corrected observations changes more. Accordingly, the repeated median hybrid filters are Lipschitz continuous with the same constant $2k+1$.

4.4 Breakdown - a worst case analysis

We can apply the concept of breakdown point to measure the amount of contamination in a single time window which makes the extracted signal value worthless, i.e. arbitrarily large or small. The finite-sample breakdown point is obtained by changing a fraction of the data points, thus getting a contaminated sample. The finite-sample breakdown point is then defined to be the smallest fraction of contaminated points such that the estimate can be any value.

When applying the median to $n = 2k+1$ data points $x_{-k}, \ldots, x_k$, the breakdown point is $(k+1)/n$ telling us that at least half of the sample needs to be 'outlying' in order to make the level approximation completely unreliable.

The breakdown point for the level estimation by each of the FMH filters is only $2/n$. The outcome of the central subfilter and of the subfilters for forward extrapolation and hence the outcome of the FMH filter can be made arbitrarily large by replacing the observation in the center and another observation which has positive weight for all predicting subfilters, e.g. the observation just at the left hand side of the center. Hence, two subsequent spikes cause the same, just somewhat dampened pattern in the extracted signal.

Breakdown of the RM affords contamination of at least k observations, while for the predictive RMH the observation in the center and at least $\lfloor k/2 \rfloor$ observations at one side of it need to be replaced. For the combined RMH, one more observation is necessary in order to destroy also the corresponding half-window median. Therefore, the breakdown points are $(\lfloor k/2 \rfloor + 1)/n$ and $(\lfloor k/2 \rfloor + 2)/n$ corresponding to replacing e.g. 6 and 7 out of $n = 21$ observations, respectively. For the predictive RMMH, at least $\lfloor k/2 \rfloor$ observations both at the left and at the right hand side of the center need to be replaced to destroy both extrapolating subfilters, i.e. the breakdown point is $2\lfloor k/2 \rfloor/n$ meaning that e.g. 10 bad observations are necessary to make the PRMMH break down if $n \in \{21, 23\}$. For the combined RMMH, destruction of both subfilters to one side affords replacement of $\lfloor k/2 \rfloor + 1$ observations there, and destruction of the linear extrapolating filter on the other side needs another $\lfloor k/2 \rfloor$ contaminated observations, summing up to $2\lfloor k/2 \rfloor + 1$ out of n, which is 11 both for $n = 21$ and $n = 23$, i.e. the breakdown point is the same as for the median or slightly smaller. Table 2 summarizes these results.

Table 2: Breakdown points in a single time window.

SM	RM	FMH	PRMH	CRMH	PRMMH	CRMMH
$\frac{k+1}{n}$	$\frac{k}{n}$	$\frac{2}{n}$	$\frac{\lfloor k/2 \rfloor + 1}{n}$	$\frac{\lfloor k/2 \rfloor + 2}{n}$	$\frac{2\lfloor k/2 \rfloor}{n}$	$\frac{2\lfloor k/2 \rfloor + 1}{n}$

5 Simulation experiments

We perform some Monte Carlo experiments for a comparison of the filter procedures in basic data situations and in the presence of observational noise. We use the components model (1) with $\mu_t = \mu_0 + t\beta$ generating the observational noise from an autoregressive model of order one, AR(1), $u_t = \phi u_{t-1} + \epsilon_t$, where the innovations ϵ_t form Gaussian white noise with mean zero and variance σ^2. AR(1) models are a convenient choice for modelling autocorrelations. Because of location and scale equivariance we set $\mu_0 = 0$ and $\sigma^2 = 1$ without loss of generality. The results for the RM, the PFMH and the PRMH do neither depend on the choice of β since these methods are trend invariant.

The suitable choice of the window width $n = 2k + 1$ depends on the application, i.e. on the situations a filtering procedure needs to handle. For removing patches of subsequent outliers we must choose k sufficiently large, while upper limits for k are imposed by the duration of periods in which trends can be assumed to be approximately linear and by the admissible time delay. We choose a window width of $n = 2k + 1 = 21$ and investigate the attenuation of Gaussian noise, the preservation of level shifts and the removal of outlier patches.

5.1 Efficiency

Firstly we compare the attenuation of Gaussian noise. All methods are unbiased then because of symmetry. We simulate 50000 time windows for each slope $\beta = 0, 0.05, \ldots, 0.5$ and correlations $\rho \in \{0, 0.6\}$ in order to compare the efficiencies of the filters as measured by the percentage MSE relative to the arithmetic mean, see Figure 3.

The standard median looses a lot of efficiency with an increasing slope, while the RM is almost as efficient as the median for a constant signal throughout. The usage of the one-sided arithmetic means or medians increases the efficiencies of the combined hybrid filters substantially for a nearly constant signal, but this gain gets lost with increasing slope as the filters become dominated by the subfilters extrapolating a linear trend. The combined and the predictive versions become nearly undistinguishable for a moderate trend with $\beta = 0.2$ in case of zero autocorrelations, while positive autocorrelations slow this decrease down. Each of the RMMH filters is more efficient than the corresponding FMH filter, while the corresponding RMH filter is slightly less efficient. Positive autocorrelations increase the relative efficiencies of the (repeated) median based filters substantially.

Figure 3 about here

5.2 Preservation of shifts

Level shifts need to be localized and tracked as exactly as possible. In order to investigate the preservation of a shift within a linear trend we simulate signals with $\beta \in \{-0.5, 0, 0.5\}$ and generate data situations which mimic the intrusion of a shift into the window replacing an increasing number $1, 2, \ldots, 10$ of observations at the end of the window by additive outliers of size $s \in \{2, 4, \ldots, 10\}$. For each setting we simulate 2000 windows and calculate the bias, the standard deviation and the root of the mean square error RMSE for all methods. We find larger outliers to have stronger effects and compare the maximal bias, standard deviation and RMSE for a fixed number of outliers, i.e. observations affected by the shift.

Figure 4 depicts the RMSE for a constant signal ($\beta = 0$) and for a steep upward or downward trend ($\beta = \pm 0.5$). Like in most of the following comparisons the RMSE is dominated by the bias, while the standard deviations are much less increasing with an increasing number of outliers. We restrict to the RMSE therefore. If the shift and the trend have the same direction, e.g. if both are upwards, the RM is the only filter to become strongly biased, namely when more than 5 (24%) observations are affected. The median and the hybrid methods perform very well then, the FMH and the RMH filters being even better than the median in case of a constant signal. This picture changes substantially if the shift and the trend have opposite directions. As outlined in section 4.2, the median smooths the shift then. The CRMMH resists up to four (19%), the CRMH and the PRMMH up to six (29%) and the CFMH up to 7 (33%) affected observations. Only the PFMH and the PRMH do not smooth a shift at all as for a constant signal since they are trend invariant. In the case of positive autocorrelations $\phi = 0.6$, we find an increase of variance for all methods, but the results are essentially the same otherwise.

Figure 4 about here

5.3 Removal of patchy impulsive noise

We investigate the ability of the methods to remove impulsive noise (outliers). In applications like intensive care with vital sign time series we find irrelevant patches of several subsequent outliers of similar size, which should not influence the outcome of the filtering, and sometimes even more than one outlier patch occurs within a short time period.

For some of the methods the location of the outliers is essential for the effect they have. The hybrid methods are to be expected to perform well when all outliers are located at the same side of the window as then only one and two subfilters respectively are affected. The FMH and RMH filters are influenced most if the central observation is outlying. Since the window is moved through the data an outlier patch will be found at any location in the window at some time point. We consider several settings with either one outlier patch in the center of the window or with two separated outlier patches. The situation with one outlier patch at the start or at the end of the window has been treated implicitly in the previous subsection. In each situation we replace an increasing number $1, 2, \ldots, 10$ of observations by additive outliers of size $s \in \{2, 4, \ldots, 10\}$ and calculate the RMSE, the bias and the standard deviation from 2000 simulation runs for each setting.

Figure 4 reports the RMSE's for one outlier patch next to the center of the window, where we replace the observations at time points $0, 1, 2, \ldots, 9$ by positive outliers. We find this situation to be a worst case for most of the methods. The RMSE's are again mainly due to bias, only the PRMH shows an increase of variance for 3 to 5 outliers. The PFMH is strongly biased already in case of 2 outliers, while the PRMH resists them much better. The performance of the other methods depends on the underlying slope. Two positive outliers within an upward trend damage the CFMH considerably, while the CRMH resists 2 (10%) outliers in case of a trend and 5 (24%) outliers in case of a constant signal satisfactorily. The RMMH filters perform very similarly to the median showing a weakly increasing bias. The best method in case of a trend is the RM, which resists about 6 (29%) outliers in the center.

When moving the time window further getting the outlier patch right in the center of the window at time points $0, 1, -1, 2, -2, \ldots, 5$ (not shown here), the RMH filters perform somewhat better than before, while the PRMMH is strongly biased in case of more than 5 (25%) outliers even in case of a constant signal. In a trend period, the RM is again the most resistent method.

The repeated median hybrid filters also resist two patches of additive outliers within half a window width distance at $t = 0, 10, -1, 9, -2, 8, -3, 7, -4, 6$ much better than the FMH filters, see Figure 5. If both patches have the same sign, the PRMH resists up to four outliers, which is twice as much as the FMH filters. The other filters perform very well in case of a constant signal, but the RM is the only method to remove more than six positive outliers from a negative trend. If the patches have different sign (not shown here), the PFMH is strongly affected already by two outliers, just like the CFMH in case of a trend.

Regarding situations with an outlier patch at each end of the window we replace the observations at time points $-10, 10, -9, 9, \ldots, -6, 6$ by additive outliers, see Figure 5. Outliers at both ends of the window have a smaller effect on the hybrid filters than outliers in the center. If both patches have the same sign, it needs more than three outliers for the PFMH and more than seven for the PRMH and the PRMMH to become considerably affected, which is similar to the median in a trend period. The CRMMH improves on the median as it accommodates well up to eight outliers even within a trend. The RM and the CRMH even resist up to almost 50% outliers without being affected a lot. Outlier patches with different sign cause much smaller problems here as the bias effects of the outliers nearly cancel each other out.

Figure 5 about here

6 Application to time series

We now apply the procedures to time series with an underlying complex signal choosing the window width again as $n = 21$. At the start and the end of the series we append the first and the last value respectively for signal extraction except for the RM, for which we use the regression lines in the last first and the last window.

6.1 Simulated time series

We simulate a time series of length 300 with an underlying signal containing constant periods as well as trend periods and three shifts, which is all overlaid by N(0,1) white noise, see Figure 6. 5% of the observations have been replaced by outliers of size -10, which are positioned as three isolated, three pairs and two triples of outliers, which have been inserted at time points chosen at random. The repeated median preserves the shift within the constant signal worse, but those within trends better than the median, and it is smoother during the trend periods. The FMH and the PRMH are more variable than these, but they preserve the shifts much better. The PFMH shows similar spikes as the time series when these have a length of at least two. The PRMH is only mildly affected by these and we find it to perform similarly to a standard median with a suitably chosen short window width. However, for the latter the choice of the window width is more critical as it tracks outlier patches with a length more than half this width, while the PRMH accommodates them largely. The CRMH and the PRMMH preserve the shifts and the extremes better than the RM but they are more volatile.

Figure 6 about here

We use this example for getting information on the number S of moves needed for updating the repeated median when using the new algorithm which only needs linear space. Figure 7 illustrates that S increases linearly with increasing window width for a steady state or a trend period, but quadratically when a shift or a local extreme occur. This points at a running time of $O(n \log n)$, but in a worst case, e.g. at a shift, it might be $O(n^2 \log n)$.

Figure 7 about here

6.2 Real time series

We analyze two time series representing arterial blood pressure, see Figures 8 and 9. Again the repeated median (not shown here) results in the smoothest signal, but like the median it underestimates the extremes. The PFMH and the PRMH preserve the local extremes and track long fluctuations, and the PRMH is not affected by short patches. The PRMMH and the CRMH are in between the PFMH, the median and the repeated median. The CRMH performs very well in the second example.

Figure 8 about here

Figure 9 about here

7 Summary

As expected none of the filters investigated here is overall optimal. Standard median filters are highly robust and preserve edges in case of a constant signal, but they worsen considerably in trend periods. Repeated median filters are not affected by trends and attenuate both Gaussian and spiky noise well. However, they smooths shifts and extremes considerably. Predictive FMH filters are also not affected by trends and preserve shifts better than the median, but they are neither efficient for Gaussian noise nor robust. Combined FMH filters attenuate Gaussian noise more efficiently than predictive FMH filters in case of a constant signal and track shifts better than the median, but worse than the predictive FMH.

We prefer repeated median hybrid filters to the FMH filters since they provide the same benefits and are considerably more robust as they accommodate a substantially larger number of spikes. Update algorithms allow fast computation of these robust filters. The predictive RMH filter is interesting as it is not affected by trends and can preserve shifts exactly. Its major drawback might be its low efficiency, which might be improved upon in a second step by applying a linear filter to its outcome. The combined RMMH shows a performance close to the median offering similar robustness properties and slightly better performance in trend periods, but being less efficient. The combined RMH and the predictive RMMH can be seen as compromises between these extremes. They offer larger efficiency and higher robustness than the predictive RMH and preserve shifts better than the median. Although the predictive RMMH has the better analytical properties since more spikes are needed to influence its results and since it can preserve a shift irrespective of the directions of the shift and an underlying trend, we have found the combined RMH to perform better in practice.

Acknowledgements

The financial support of the Deutsche Forschungsgemeinschaft (SFB 475, "Reduction of complexity in multivariate data structures") is gratefully acknowledged. This work was initiated while the first author visited the University College London supported by the European Community's Human Potential Programme under contract HPRN-CT-2000-00100 (DYNSTOCH). The authors thank Eleni Mitropoulou for additional data analysis.

References

Astola, J., Heinonen, P., Neuvo, Y., 1989. Linear median hybrid filters. IEEE Transactions on Circuits and Systems 36, 1430–1438.

Bernholt, T., Fried, R., 2003. Computing the update of the repeated median regression line in linear time. Information Processing Letters 88, 111–117.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., 1990. Introduction to algorithms. MIT Press, Cambridge, Massachusetts, and McGraw-Hill Book Company, New York.

Davies, P.L. 1993. Aspects of robust linear regression. Ann. Statist. 21, 1843–1899.

Davies, P., Fried, R., Gather, U., 2004. Robust signal extraction for on-line monitoring data. J. Statist. Plann. Inference, to appear.

Heinonen, P., Kalli, S., Turjanmaa, V., Neuvo, Y., 1985. Generalized median filters for biological signal processing. Proceedings of the Seventh European Conference on Circuit Theory and Design, 283–386.

Heinonen, P., Neuvo, Y., 1987. FIR-median hybrid filters. IEEE Transactions on Acoustics, Speech, and Signal Processing 35, 832–838.

Heinonen, P., Neuvo, Y., 1988. FIR-median hybrid filters with predictive FIR substructures. IEEE Transactions on Acoustics, Speech, and Signal Processing 36, 892–899.

Kalli, S., Heinonen, P., Turjanmaa, V., Saranummi, N., 1985a. Identifying patterns and profiles in 24-hour blood pressure recordings. Proceedings of the Fifth International Symposium on Ambulatory Monitoring, 113–118.

Kalli, S., Heinonen, P., Turjanmaa, V., Saranummi, N., 1985b. Decomposition of Continuous Long-Term Blood Pressure Signals into Root Components. *IEEE Computer Society*, 307–310.

Nieminem, A., Neuvo, Y., Mitra, U., 1989. Algorithms for real-time trend detection. Signal Processing 18, 1–15.

Siegel, A.F., 1982. Robust regression using repeated medians. Biometrika 68, 242–244.

Stein, A., Werman, M., 1992. Finding the repeated median regression line. In: SODA: ACM-SIAM Symposium on discrete algorithms, 409–413.

Tukey, J.W., 1977. Exploratory Data Analysis. Addison-Wesley, Reading, Mass., (preliminary ed. 1971).

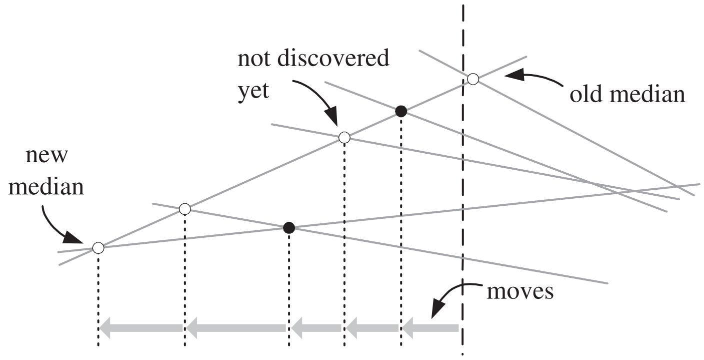


Figure 1: After a new line is inserted, the imaginary vertical line is moved from the old median towards the new median. The black intersection points are visible to the algorithm at this point in time and stored in the heap. The white intersection points become visible if the two incident lines become neighbored in the permutation.

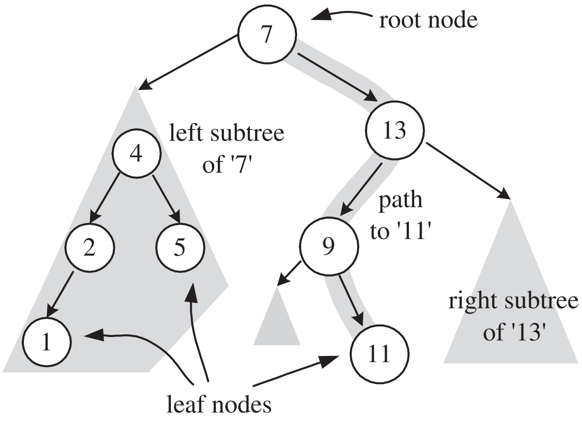


Figure 2: Each data item is stored in a node of the binary search tree. Each node has pointers to a left and/or right child node. If neither left nor right child exists, the node is called a leaf. The depth of the tree is the number of nodes on the longest path from the root to a leaf.

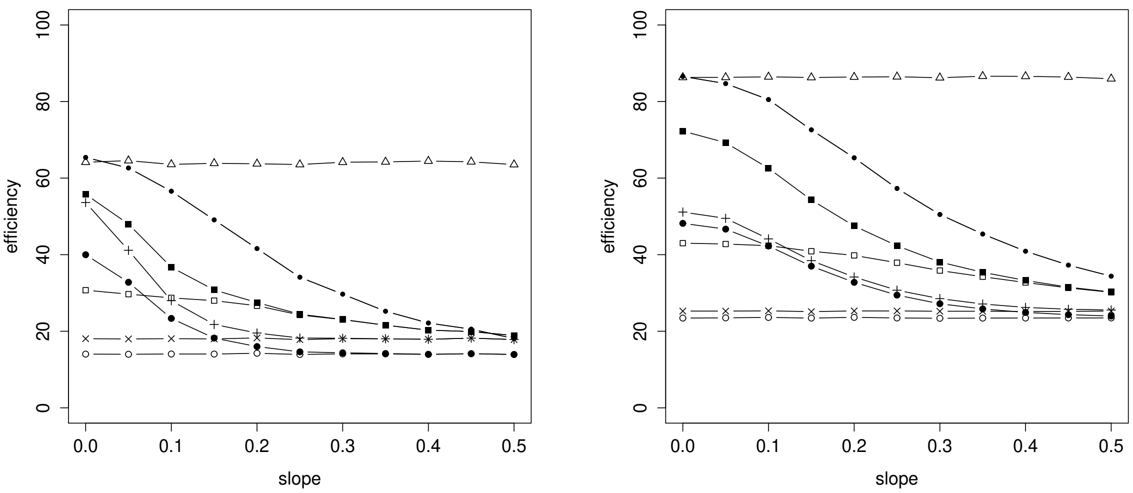


Figure 3: Relative efficiencies for Gaussian noise, autocorrelations $\rho = 0.0$ (left) and $\rho = 0.6$ (right): median ($\cdot$), RM ($\triangle$), PFMH ($\times$), CFMH ($+$), PRMH ($\circ$), CRMH ($\bullet$), PRMMH ($\square$) and CRMMH ($\blacksquare$).

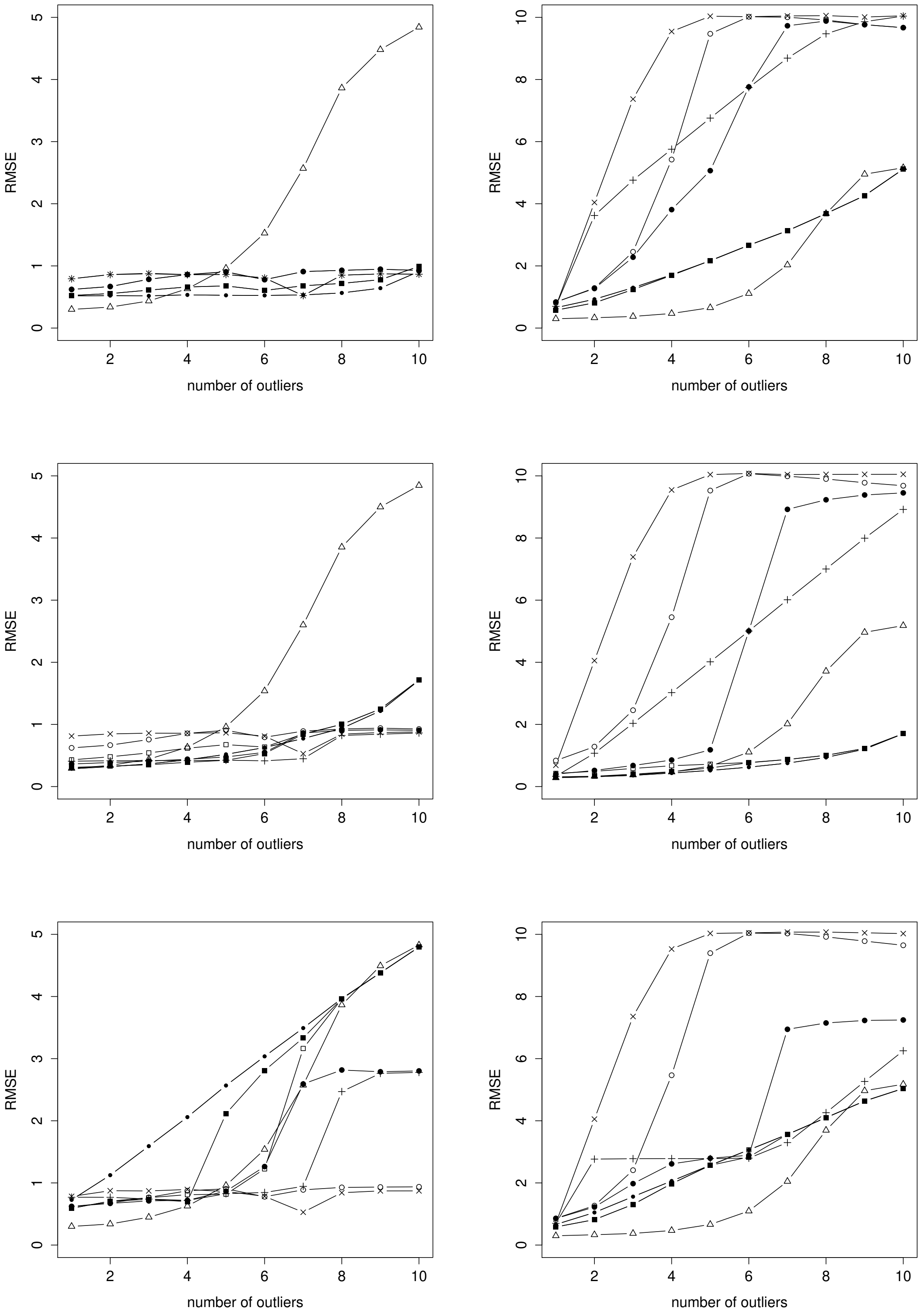


Figure 4: RMSE for the intrusion of a level shift (left) and for an outlier patch just to the right of the center (right), slope $\beta = 0.5$ (top), $\beta = 0.0$ (center) and $\beta = -0.5$ (bottom): median ($\cdot$), RM ($\triangle$), PFMH ($\times$), CFMH ($+$), PRMH ($\circ$), CRMH ($\bullet$), PRMMH ($\square$) and CRMMH ($\blacksquare$).

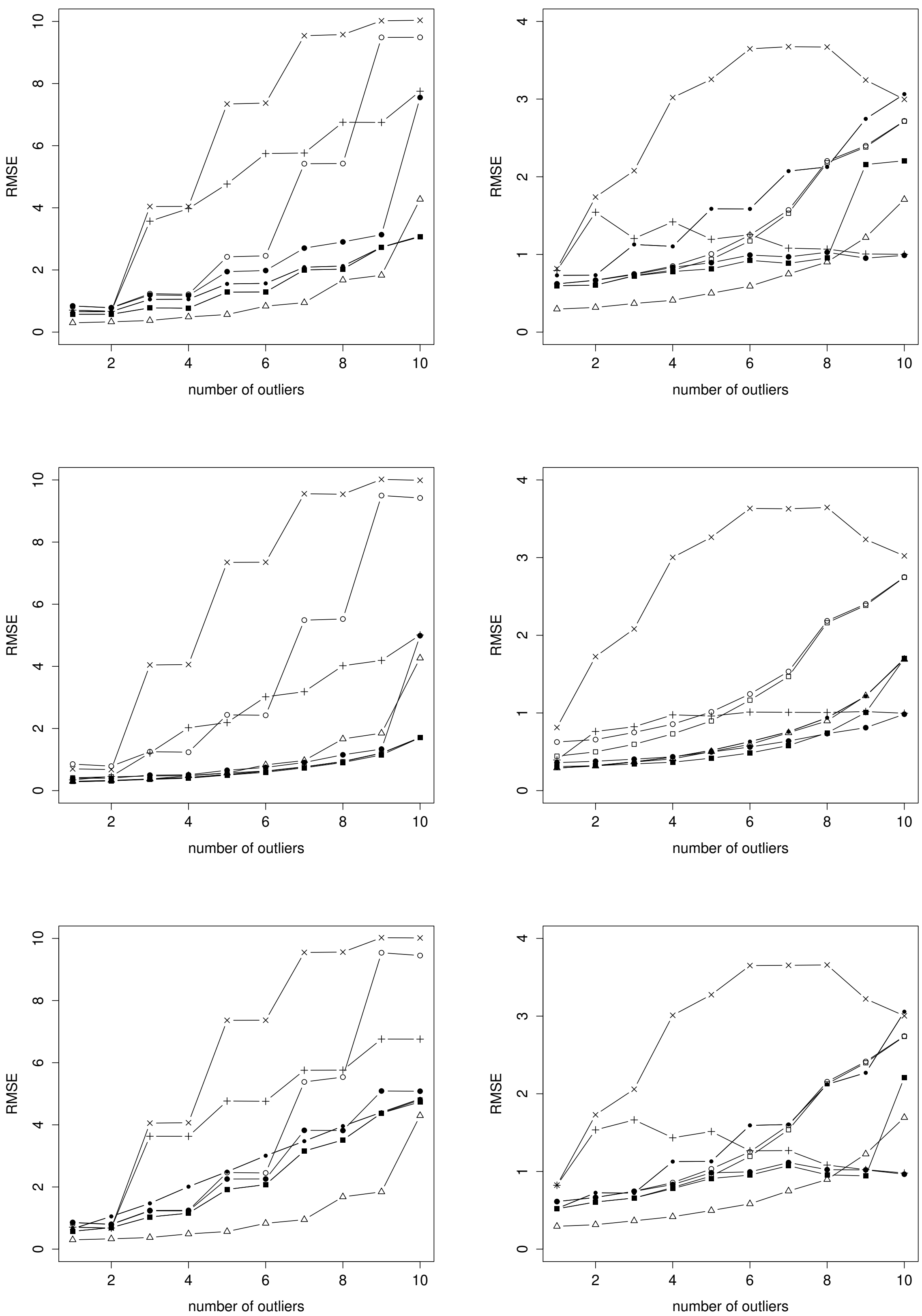


Figure 5: RMSE for positive outlier patches lagged by a half (left) and a full window width, slope $\beta = 0.5$ (top), $\beta = 0.0$ (center) and $\beta = -0.5$ (bottom): median ($\cdot$), RM ($\triangle$), PFMH ($\times$), CFMH ($+$), PRMH ($\circ$), CRMH ($\bullet$), PRMMH ($\square$) and CRMMH ($\blacksquare$).

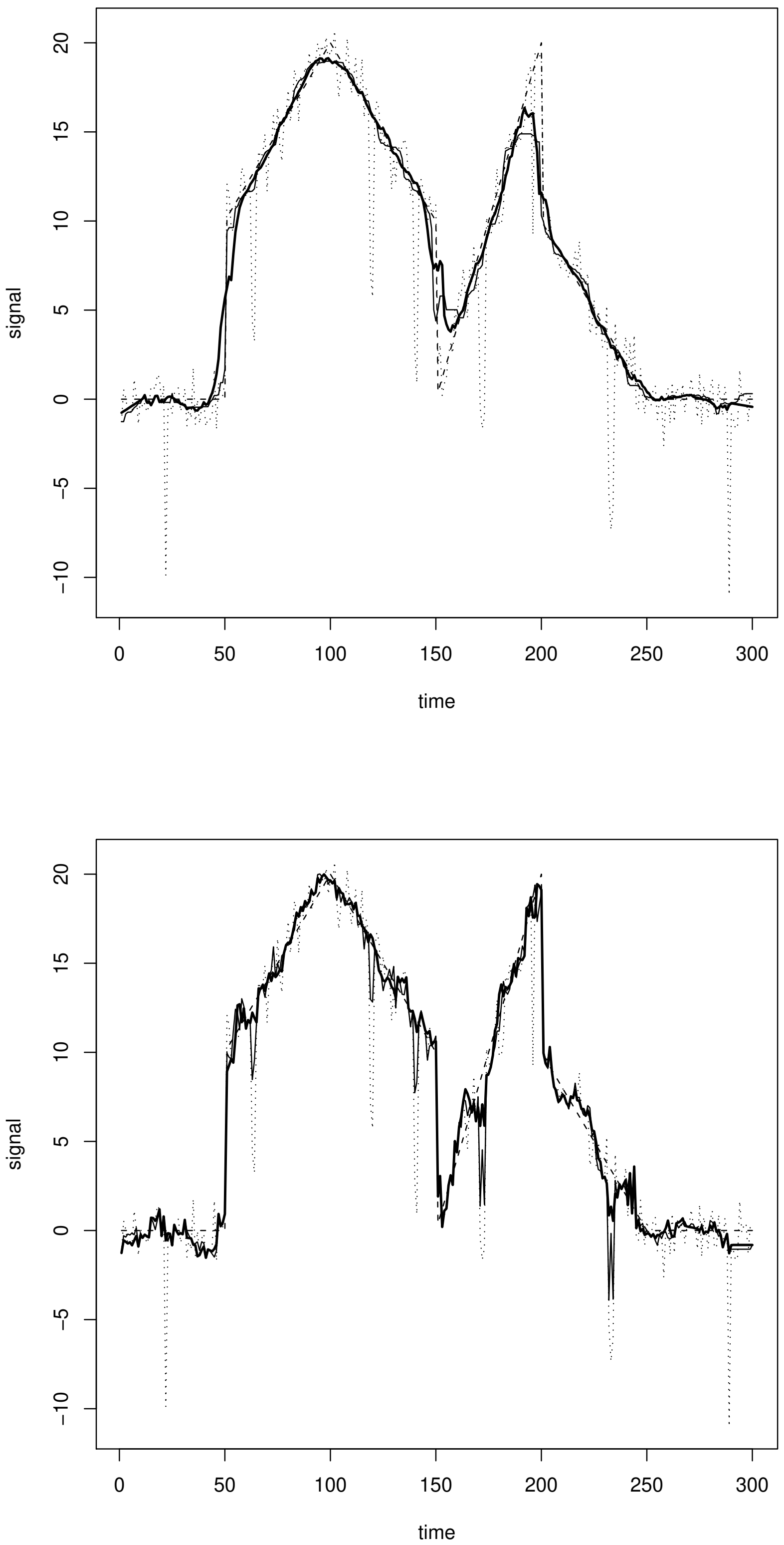
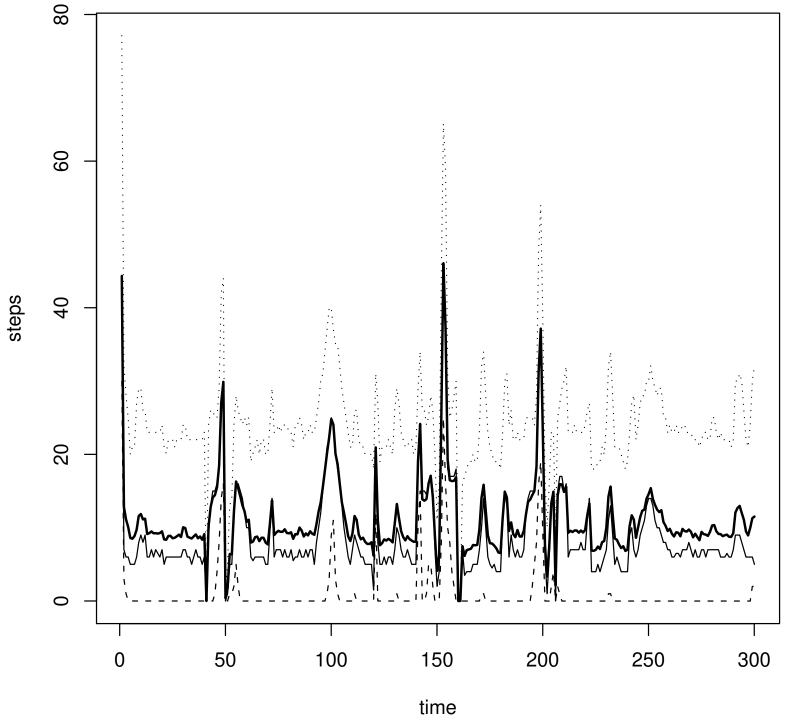


Figure 6: Simulated time series with outlier patches of length up to three (dotted) and underlying signal (dashed). Top: SM (thin solid), RM (bold solid). Bottom: PFMH (thin solid), PRMH (bold solid).



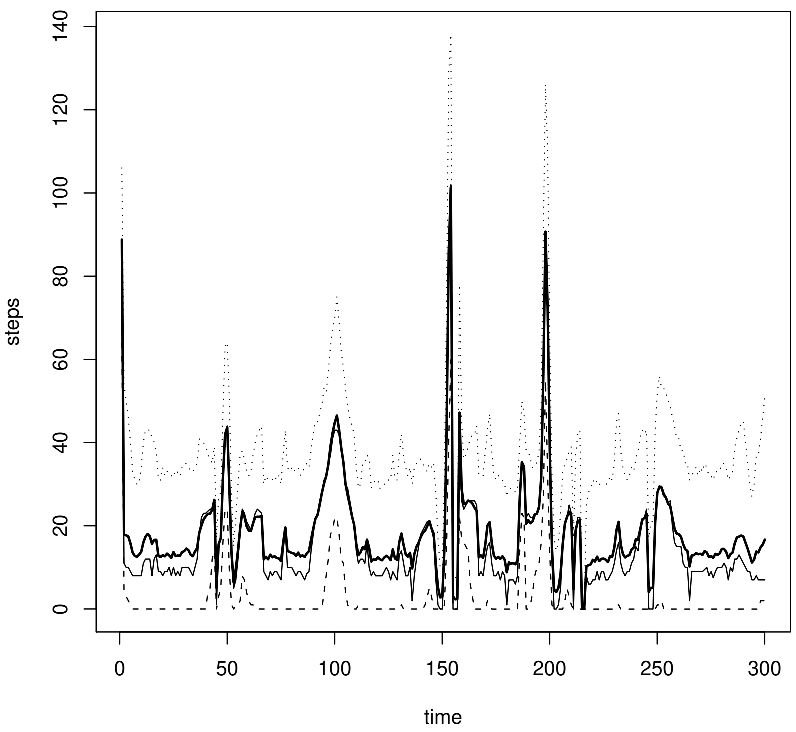
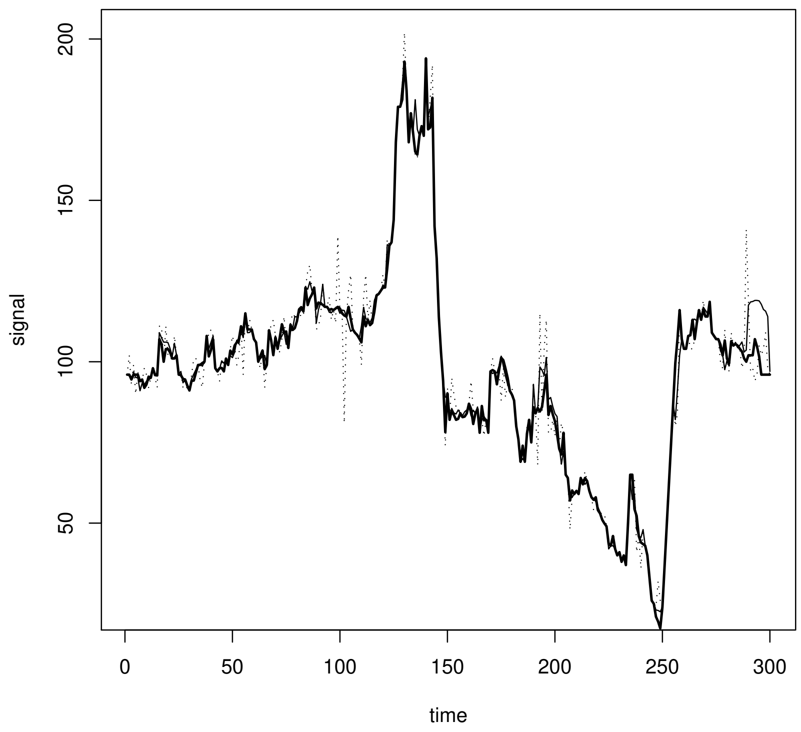


Figure 7: Number S of moves needed for updating the repeated median at each time point, window width $n = 21$ (top) and $n = 31$ (bottom), upper 10 percentage point (dotted), median (thin solid), mean (bold solid), and lower 10 percentage point (dashed).



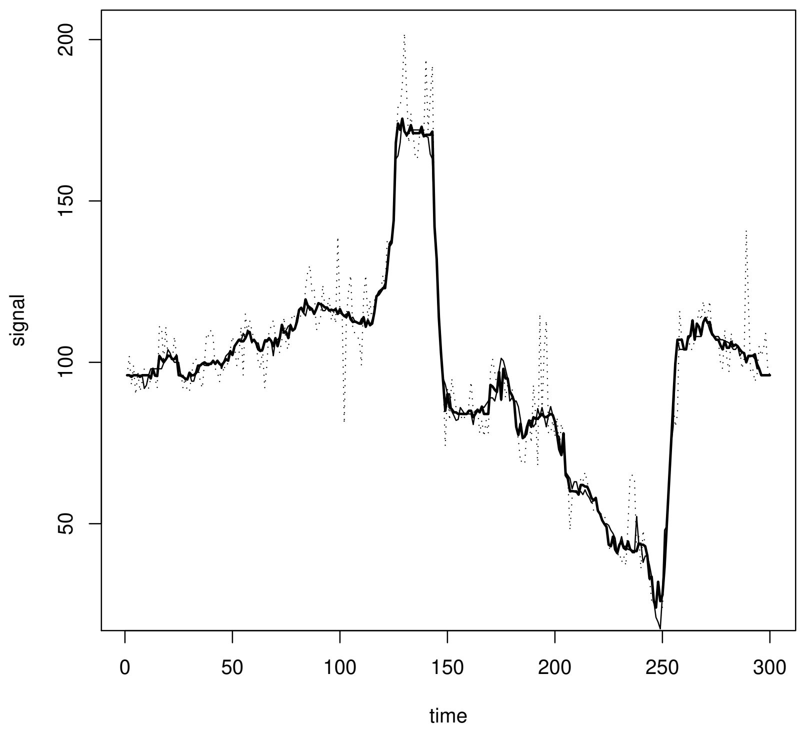


Figure 8: Time series representing arterial pressure (dotted) and extracted signals. Top: PFMH (thin solid), PRMH (bold solid). Bottom: PRMMH (thin solid), CRMH (bold solid).

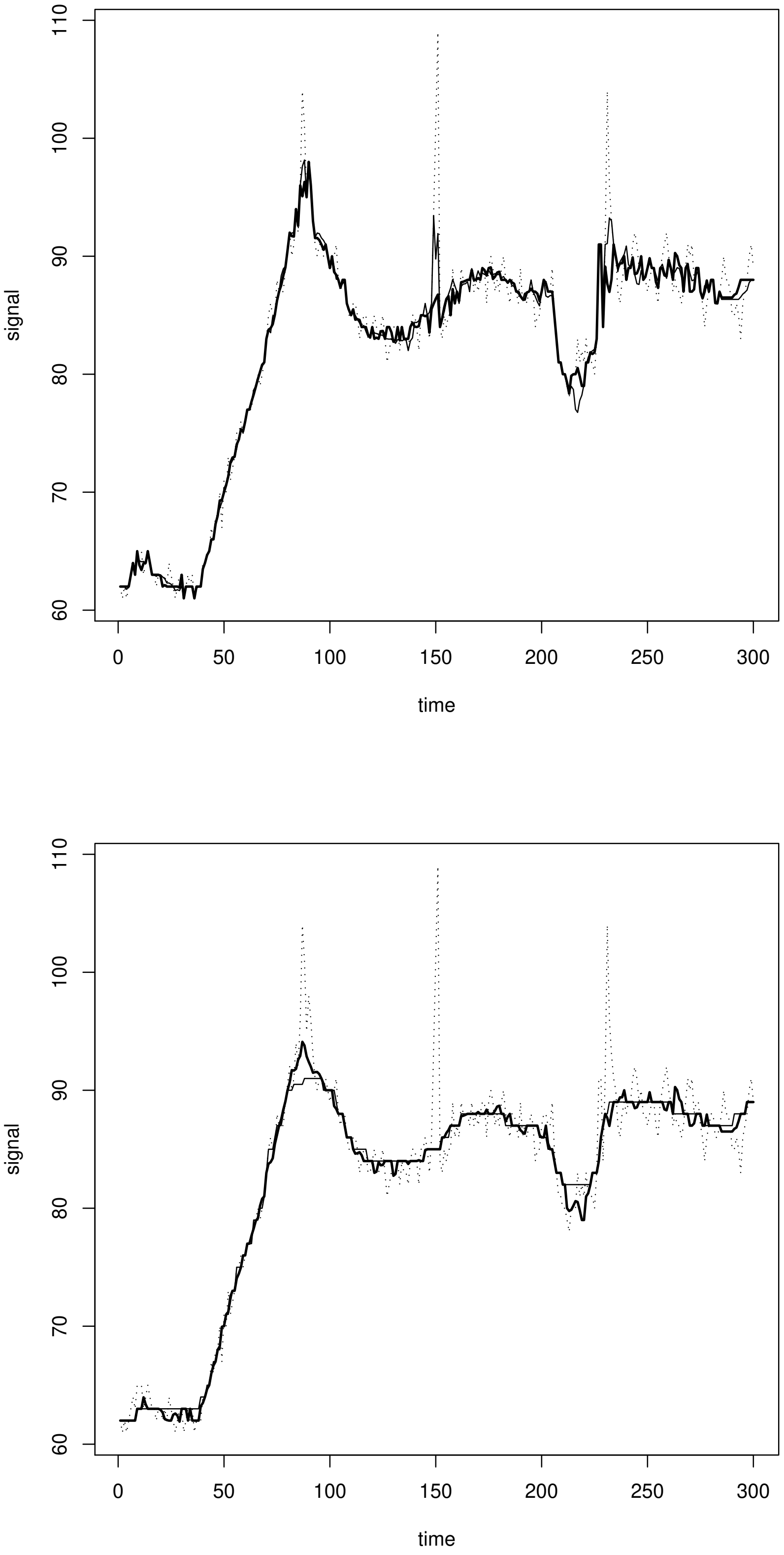


Figure 9: Time series representing arterial pressure (dotted) and extracted signals. Top: PFMH (thin solid), PRMH (bold solid). Bottom: PRMMH (thin solid), CRMH (bold solid).