Complexity Penalised M-Estimation: Fast Computation

F. Friedrich* A. Kempe† V. Liebscher‡ G. Winkler §

November 28, 2005

Keywords and Phrases: complexity penalised variational problems, Potts model, edge preserving smoothing, regularisation, timeseries, nonlinear filters, segmentation
Mathematical Subject Classification: 93E14, 62G08, 65K05, 90C39, 90C30, 90C31

Abstract

We present fast algorithms for the exact computation of estimators for time series, based on a simple variational approach. The functionals behind are complexity penalised loglikelihood- or M-functions. We emphasize optimisation simultaneously in all model parameters. The algorithms cover a broad range of estimators, including all those commonly adopted in the literature. This is illustrated by a series of examples.

1 Introduction

In this paper we present algorithms for the fast computation of complexity penalised M-estimators for time series. Complexity penalised likelihood functions appear in the literature in various contexts. We start with a brief motivation by way of example.

Penalised sums of squared deviations are classical models of the form

$$P_\gamma : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}, \quad (x, y) \longmapsto \gamma \cdot |J(x)| + \sum_{i=1}^{n} (y_i - x_i)^2, \ \gamma \geq 0. \tag{1}$$

Let us make this precise. There is a finite set $T = \{1, \ldots, n\}$ of time points. Elements of $\mathbb{R}^n$ are interpreted as time series or signals $x = (x_1, \ldots, x_n)$. For each time series x, the *set of jumps* is $J(x) = \{i = 1, \ldots, n-1 : x_i \neq x_{i+1}\}$, and their number is denoted by $|J(x)|$. The time series $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ is interpreted as measurement or data, and each x is a candidate for the representation of data subject to the (soft) restrictions imposed by the functional. An *estimate* is a signal which minimises $P_\gamma(\cdot, y)$. These estimates enjoy an optimal tradeoff between fidelity to data, measured by the sum of squares, and complexity, measured by the number of jumps.

*Institute of Biomathematics and Biometry, GSF; partially supported by DFG grant SFB 386 at the LMU München.

†Münchener Rückversicherung, München; partially supported by DFG Graduate Programme 'Applied Algorithmic Mathematics' at the TU München and DFG grant SFB 386 at the LMU München

‡University of Greifswald

§Corresponding author: G. Winkler, IBB - Institute of Biomathematics and Biometry, GSF - National Research Centre for Environment and Health, Postfach 1129, D-85758 Oberschleißheim, Germany, gwinkler@gsf.de, http://ibb.gsf.de

From a Bayesian point of view, these functionals are negative posterior loglikelihood functions with an improper prior. Minimal points of $P_\gamma(\cdot, y)$ correspond to the respective maximum a posteriori estimates.

The following reformulation is convenient. Each signal $x \in \mathbb{R}^n$ can be described by the family $\mathcal{P}$ of those maximal discrete intervals $I \subset T$ on which it is constant, and by the values $\mu_I \in \mathbb{R}$ which it takes on I. Then (1) can be rewritten as

$$\tilde{P}_\gamma : \ (\mathcal{P}, (\mu_I)_{I\in\mathcal{P}}) \longmapsto \gamma(|\mathcal{P}| - 1) + \sum_{I\in\mathcal{P}} \sum_{i\in I} (y_i - \mu_I)^2.$$

Taking minima yields

$$\begin{aligned} &\min_{(\mathcal{P},(\mu_I)_{I\in\mathcal{P}})} \tilde{P}_\gamma(\mathcal{P}, (\mu_I)_{I\in\mathcal{P}}) \\ &\quad = \quad \min_{\mathcal{P}} (\gamma(|\mathcal{P}| - 1) + \sum_{I\in\mathcal{P}} \min_{\mu_I} \sum_{i\in I} (y_i - \mu_I)^2) = \min_{\mathcal{P}} (\gamma(|\mathcal{P}| - 1) + \sum_{I\in\mathcal{P}} \sum_{i\in I} (y_i - \bar{y}_I)^2) \end{aligned}$$

with the empirical means $\bar{y}_I = (\sum_{i\in I} y_i)/|I|$. For sums of absolute deviations the mean is replaced by the median. Note that partitions in which the minima are attained consist of maximal intervals on which the signal is constant.

This is an example for a general *reduction principle*, which applies to a large class of functionals. It can be exploited whenever the minimal values μ_I on the righthand side are known. Then the original optimisation problem on $\mathbb{R}^n$ boils down to one on the finite set of partitions. Unfortunately, the cardinality of this set is still 2^{n-1} and thus grows exponentially in the sample size. In order to develop fast optimisation algorithms we will exploit ideas from dynamic programming.

One of the main problems - appearing in almost all similar situations - is the identification of the smoothing or hyper parameter γ. It should depend on the actual data and must be adapted to the aims of the concrete statistical analysis. To attack this problem, it is desired, and in fact very helpful, to have the estimates for *all* parameters γ. Our theoretical findings, originating from O. Wittich et al. (2005), paved the way to the development of an algorithm which computes all desired estimates together in time complexity better than $O(n^3)$. It is reported in Section 4 below.

Before we proceed with more general estimates, let us point out that the estimators for constant local regression are themselves of considerable interest. For example, they are especially useful for the detection of (multiple) change points in time series, see P. Bhattacharya (1994), M. Csörgö and L. Horváth (1997), H.-G. Müller (1992) or C. R. Loader (1996). Moreover, they can be used as a preprocessing step in order to partition the region of interest into homogeneous parts. Afterwards, any kind of smoothers or other filters can be applied to the single subregions. For multidimensional signals this aspect is addressed in D. Geman et al. (1987).

If we insist on a penalty proportional to the number of pieces into which a signal is decomposed there remain two directions of generalization. One is to modify the distance between signal and data: Sums of squared deviations belong to Gaussian white noise. The algorithms below apply to a much wider class of degradations. The most popular, and more robust, alternative is double exponential white noise; sums of squares are then replaced by sums of absolute deviations. The algorithms lend themselves also to M-estimators of location, which in general do not allow for a likelihood in the strict sense, and also to even more general estimators. A second possibility of generalization is to replace the constant local regressors for example by polynomials or splines, or by functions with prescribed

morphological properties like monotonicity or their number of modes. We will illustrate the indicated wide range of applications by a series of typical examples in Sections 5 and 6.

Originally, the authors developed algorithms, similar to those reported below, for functionals of the type in (1). The initiative came from data with either little or no ground truth behind, or where there was strong evidence that meaningful and interpretable estimates should be piecewise constant. Such examples, from the analysis of gene expression data (cf. Fig. 5), and from brain mapping of responses to boxcar shaped stimuli, are reported in G. Winkler et al. (2005) or A. Kempe (2004), and A. Kempe et al. (2005)

Motivated on these grounds, the authors of the present paper, jointly with others, studied the model (1) in a series of papers. V. Liebscher and G. Winkler (1999) introduced the basic scheme of the algorithms. G. Winkler and V. Liebscher (2002), A. Kempe (2004), and O. Wittich et al. (2005) discussed deterministic properties. V. Liebscher et al. (2004) embedded the model into a family of functionals including the (continuous time) Mumford-Shah functional. In A. Kempe (2004) and L. Boysen et al. (2005), statistical aspects like consistency and rates of convergence were addressed.

Functionals with a complexity penalty appear in many papers; let us mention just a few 'classical' ones. The penalty itself was introduced in R. Potts (1952) as the energy function of a spin system with finitely many states. In S. Geman and D. Geman (1984) such functionals are mentioned in the context of signals with discrete values, with focus on multidimensional 'images'; in A. Blake (1983) and A. Blake and A. Zisserman (1987) they are extreme cases of what is nowadays called Blake-Zisserman models. Complexity penalised likelihoods have developed into a standard tool in nonparametric statistics, see for example L. Györfi et al. (2002) for an account. D. Donoho (1999) studies these functionals in two dimensions restricting the class of partitions to those which consist of elements with wedge-shape, so-called wedgelets. For an up-to-date account of this circle of ideas see H. Führ et al. (2006).

Finally, we sketch briefly the plan of this paper. The general optimisation problem will be formulated in Section 2. In Section 3 we will describe an algorithm for single parameters γ, and in Section 4, we will compute minimising time series for all parameters simultaneously. In Section 5, we apply the general scheme to special functionals and data, mainly with loglikelihoods in ℓ^p. In the last Section 6, we indicate the flexibility of the present approach, and argue that it applies to a wide variety of situations. Modifications are mainly in the computation of the quantities d_I^* substituting $\sum_{i\in I}(y_i - \bar{y}_I)^2$ in the reduction principle.

2 Formulation of the Problem

We are now going to formulate the general variational problem. The key to the construction of fast algorithms is the reduction of the minimisation problem on $\mathbb{R}^n$ to one on a finite set. To this end, we describe signals $x \in \mathbb{R}^n$ in terms of segmentations, i.e. by intervals I in T on which x has characteristic properties.

To make the latter precise, we associate to each interval I a space $\mathcal{F}_I$ of functions $\mu_I : I \to \mathbb{R}$. For some applications, we adopt the usual setting from approximation theory, where $\mu_I(i) = f(t_i)$, $i \in I$, for functions f on $\mathbb{R}$ with prescribed smoothness properties, and design points $t_i \in \mathbb{R}$. The functions f may, for example, be constant like in the Introduction, polynomials of higher degree, or splines. In another class of examples, morphological properties like monotonicity, uni- or multi-modality are in the focus. One can even use templates of special shape which the estimated time series should resemble.

A *partition* of T is a collection $\mathcal{P}$ of mutually disjoint discrete intervals $I \subset T$ with union T. The set of partitions will be denoted by $\mathfrak{P}$. A pair $(\mathcal{P}, \mu)$ with

$$\mathcal{P} \in \mathfrak{P} \quad \text{and} \quad \mu = (\mu_I)_{I \in \mathcal{P}} \in \prod_{I \in \mathcal{P}} \mathcal{F}_I$$

will be called a *segmentation*, and the set of segmentations will be denoted by $\mathfrak{S}$. In these terms, we define functionals

$$H_\gamma : \mathfrak{S} \times \mathbb{R}^n \longrightarrow \mathbb{R}, \quad ((\mathcal{P}, \mu), y) \longmapsto \gamma \cdot (|\mathcal{P}| - 1) + D((\mathcal{P}, \mu), y), \tag{2}$$

where $\gamma \geq 0$ is the control parameter of the penalty. Concerning D, we will only assume that it is the sum of independent contributions from single intervals. Hence we consider data terms of the form

$$D((\mathcal{P}, \mu), y) = \sum_{I \in \mathcal{P}} d_I(y_I, \mu_I), \;\; \mathcal{P} \in \mathfrak{P}, \;\; \mu = (\mu_I)_{I \in \mathcal{P}} \in \prod_{I \in \mathcal{P}} \mathcal{F}_I, \;\; y \in \mathbb{R}^n, \tag{3}$$

with $y_I = (y_i)_{i \in I} \in \mathbb{R}^I$ and functions $d_I : \mathbb{R}^I \times \mathcal{F}_I \to \mathbb{R}$. Let us point out once more that there are two main aspects inherent in these models: morphological or smoothness properties of the local regression, made precise by the choice of the function spaces $\mathcal{F}_I$, and the local distances d_I between data and representations. We illustrate this by way of two simple examples.

Although rather simple, piecewise constant regression in the first example is important.

Example 1 Suppose that each function space $\mathcal{F}_I$ consists of the constant functions. If constant functions are identified with their unique value $\mu_I \in \mathbb{R}$. For sums of squared deviations, one gets $d_I(y_I, \mu_I) = \sum_{i \in I} (y_i - \mu_I)^2$. With the the same spaces $\mathcal{F}_I$, but the sum $d_I(y_I, \mu_I) = \sum_{i \in I} |y_i - \mu_I|$ of absolute deviations, estimation becomes more robust.

The second example addresses basic morphological features.

Example 2 Let $\mathcal{F}_I$ be the union of all de- or increasing time series on I. This corresponds to locally monotone regression. Locally, it is either antitone or isotone, depending on the better fit to data. Unless x has monotonously increasing regions with sudden jumps down (or vice versa for decreasing shape), the penalty $|\mathcal{P}|-1$ measures the number of local modes, cf. Fig. 1. Here we assumed tacitly, that the intervals are maximal in the sense that if the signal increases (decreases) on two adjacent intervals then it does not increase (decrease) on their union. It turns out that minimal points fulfil this property. The computation of estimates in this case and for explicitly penalised modes will both be indicated in Section 6.

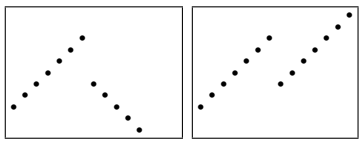

Figure 1: Sudden jump down with one and two modes near the jump

Throughout the paper, we will assume that the contributions of single intervals can be minimised separately, and hence we require:

Hypothesis 1 *For each interval $I \subset T$, the function $d_I(y_I, \cdot) : \mathcal{F}_I \to \mathbb{R}$, $y_I \in \mathbb{R}^I$, attains a minimum. A function $\mu_I^* \in \mathcal{F}_I$ in which $d_I(y_I, \cdot)$ is minimal, as well as the value $d_I^* = d_I(y_I, \mu_I^*)$, are stored.*

Under these hypotheses, the formulations (2) and (3) of the variational problem pave the way to a considerable simplification. In fact, the minimisation of (2) can be split into the minimisation in μ for each of the partitions $\mathcal{P}$, followed by the minimisation over all partitions. Formally, this reads

$$\min_{(\mathcal{P},\mu)\in\mathfrak{S}} H_\gamma((\mathcal{P},\mu),y) = \min_{\mathcal{P}\in\mathfrak{P}} \Big(\gamma(|\mathcal{P}|-1) + \sum_{I\in\mathcal{P}} d_I^*\Big). \tag{4}$$

We will call this identity the *reduction principle*. It is very much at the heart of the algorithms to be developed below. One reads off from the righthand side that under Hypothesis 1 the minima exist.

Due to the reduction principle and given the quantities d_I^*, the following optimisation problem remains to be solved:

$$\text{minimise} \quad \tilde{H}_\gamma : \mathfrak{P} \longrightarrow \mathbb{R}, \;\; \mathcal{P} \longmapsto \gamma(|\mathcal{P}|-1) + \sum_{I\in\mathcal{P}} d_I^* \tag{5}$$

After an optimal partition $\mathcal{P}^*$ for this functional is determined, an optimal segmentation for (2) is obtained piecing together the corresponding optimal signal segments μ_I^*, $I \in \mathcal{P}^*$.

3 The Basic Algorithm

Let, for the present, data y and a parameter γ be given. To formulate the recursive algorithms below, we must restrict partitions and segmentations to subintervals of T. Let left and right bounds $l, r \in \mathbb{N}$ with $1 \le l \le r \le n$ be given and let us denote discrete intervals $\{l, \dots, r\}$ by $[l, r]$. The sets of partitions $\mathfrak{P}(r)$ and segmentations $\mathfrak{S}(r)$ on intervals $[1, r]$, $r \ge 1$, are defined in the same way as those on T. Let us further introduce the *Bellman functions*

$$B(r) = \inf_{(\mathcal{P},\mu)\in\mathfrak{S}(r)} \Big(\gamma(|\mathcal{P}|-1) + \sum_{I\in\mathcal{P}} d_I^*\Big), \;\; r \ge 1.$$

Plainly, $B(n)$ is the minimal value of (2). Now we sort the partitions according to their rightmost interval and set

$$\mathfrak{P}(l,r) = \{\mathcal{P} \in \mathfrak{P}(r) : [l,r] \in \mathcal{P}\}, \;\; 1 \le l \le r \le n$$

Then $\mathfrak{P}(r) = \bigcup_{j=1}^{r} \mathfrak{P}(j,r)$, and therefore

$$B(r) = \min_{1\le j\le r} \min_{\mathcal{P}\in\mathfrak{P}(j,r)} \gamma(|\mathcal{P}|-1) + \sum_{I\in\mathcal{P}} d_I^*. \tag{6}$$

This suggests the following fundamental recursion formula.

Lemma 1 *Let $B(0) = -\gamma$. Then*

$$B(r) = \min_{1\le j\le r} B(j-1) + \gamma + d_{[j,r]}^*, \;\; r \ge 1. \tag{7}$$

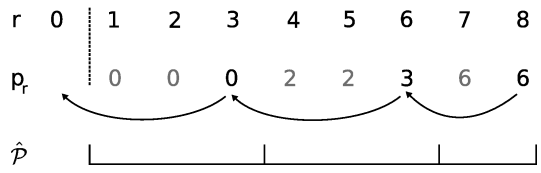

Figure 2: Data structure for one-dimensional partitions used in Algorithm 1

Proof. Choose $\mathcal{P} \in \mathfrak{P}(j, r)$, $j > 1$. Then $\mathcal{P} = \mathcal{Q} \cup \{[j, r]\}$ with a partition $\mathcal{Q} \in \mathfrak{S}(j-1)$. Therefore we have the decomposition

$$\begin{aligned} & \gamma(|\mathcal{P}| - 1) + \sum_{I \in \mathcal{P}} d_I(y_I, \mu_I) \\ = \quad & \Big(\gamma(|\mathcal{Q}| - 1) + \sum_{I \in \mathcal{Q}} d_I(y_I, \mu_I)\Big) + \Big(\gamma + d_{[j,k]}(y_{[j,k]}, \mu_{[j,k]})\Big). \end{aligned}$$

Taking minima yields the assertion. □

The previous lemma allows one to employ a dynamic programming immediately. The data structure used for storing one-dimensional partitions is an array p with length n. At position $1 \leq r \leq n$, the array p contains a best previous position:

$$p_r = \operatorname*{argmin}_{0 \leq l < r} (B(l) + \gamma + d^*_{[l+1,r]}).$$

The algorithm consists of two procedures and the main part. To avoid an overhead of technical details, we use 'global parameters' in the following pseudo-code representation of the algorithms. In a modern computer language these globally accessible variables would typically be wrapped in some composite type such as a 'record', 'class' or 'object' used together with a pointer mechanism that locates a potential repository for large data. Dynamic values as - for example - the distance function d^* or the mean values μ_I ($I \subset \{1, \ldots, n\}$) would be realised by (type bound) procedures (methods).

The first procedure delivers a best partition:

Procedure `FindBestPartition`$(\gamma \in \mathbb{R}) \in \mathbb{N}^n$

global : data length $n \in \mathbb{N}$, distance function $d^*_{[\cdot,\cdot]} \in \mathbb{R}^n \times \mathbb{R}^n$
output: partition stored in $p \in \mathbb{N}^n$
local : left and right interval bounds: $l, r \in \mathbb{N}$; Bellman values $B \in \mathbb{R}^n$; temporary $b \in \mathbb{R}$

```
begin
    B_0 := −γ (* definition *) ;
    for r ← 1 to n do
        B_r ← ∞;
        for l ← 1 to r do
            b ← B_{l−1} + γ + d*_{[l,r]};
            if b ≤ B_r then
                B_r ← b (* best value at right bound r *) ;
                p_r ← l − 1 (* best left bound at right bound r *);
            end
        end
    end
    return p;
end
```

The second procedure complements the partition by the values on intervals and delivers a segmentation:

Procedure `SegmentationFromPartition`$(p \in \mathbb{N}^n) \in \mathbb{R}^n$

global : data length $n \in \mathbb{N}$, local approximations $\mu^*_I \in \mathcal{F}_I$ for all intervals $I \subset \{1, \ldots, n\}$
output: approximation $y \in \mathbb{R}^n$;
local : left and right interval bounds: $l, r \in \mathbb{N}$; temporary $t \in \mathbb{N}$

```
begin
    r ← n; l ← p_r;
    while r > 0 do
        for t ← l + 1 to r do
            y_t ← μ_{[l+1,r]}(t);
        end
        r ← l; l ← p_r;
    end
    return y;
end
```

Both procedures are combined in the algorithm:

Algorithm 1: Minimisation of H_γ for fixed $\gamma \geq 0$

global : data length n, distances $d^*_{[l,r]} \in \mathbb{R}$, local approximations $\mu^*_{[l,r]} \in \mathcal{F}_{[l,r]}$ (for all $1 \leq l \leq r \leq n$)
input : parameter $\gamma \geq 0$
output: minimiser $\hat{y} \in \mathbb{R}^n$; partition stored in $p \in \mathbb{N}^n$
begin
    `--- minimisation ---`
    $p \leftarrow$ `FindBestPartition`(γ);
    `--- reconstruction ---`
    $\hat{y} \leftarrow$ `SegmentationFromPartition`(p);
end

Remark 1 Algorithm 1 returns the minimising partition that in each single recursion step chooses the largest possible interval. An algorithm that returns the minimising partition with the least number of intervals can also be obtained with minor modifications of Algorithm 1.

This algorithm returns the desired result.

Theorem 2 *Under Hypothesis 1, the Algorithm 1 terminates and returns a partition of T which minimises the functional (2). If evaluations of the functions $\mu_{[l,r]}$ $(1 \leq l \leq r \leq n)$ at time points $l \leq x \leq r$ take $\mathrm{O}(n)$ time and $d^*_{[l,r]}$ can be derived in $\mathrm{O}(1)$ time, then the algorithm works in time complexity $\mathrm{O}(n^2)$. The spatial complexity of the algorithm is $\mathrm{O}(n)$.*

Proof. The minimisation part of the algorithm consists of two nested finite `for`-loops and therefore terminates. Since by construction $p_r < r$ for all $1 \leq r \leq n$, the reconstruction part also terminates and therefore the algorithm delivers an output in finite time. Denote the resulting partition by $\mathcal{P}$, and assume that there is a partition $\mathcal{Q}$ of T with $\tilde{H}_\gamma(\mathcal{Q}) < \tilde{H}_\gamma(\mathcal{P})$. Then there is a least time index r such that $\mathcal{P}$ and $\mathcal{Q}$ coincide on $[r+1, n]$ but the last intervals $[p, r]$ and $[q, r]$, respectively, are different. By construction we have $p = q_r$. Let now $\mathcal{R}$ be the collection of the common intervals to the right of r. Then $\mathcal{P}$ and $\mathcal{Q}$ are disjoint unions $\mathcal{P}_r \cup \mathcal{R}$ and $\mathcal{Q}_r \cup \mathcal{R}$ with partitions $\mathcal{P}_r$ and $\mathcal{Q}_r$ of $I_r = \{1, \ldots, r\}$. The construction of $\mathcal{P}$ implies with $C = \sum_{I \in \mathcal{R}} d^*_I + \gamma|\mathcal{R}|$ that

$$B(p-1) + \gamma + d^*_{[p,r]} + C = \tilde{H}_\gamma(\mathcal{P}) > \tilde{H}_\gamma(\mathcal{Q}) \geq B(q-1) + \gamma + d^*_{[q,r]} + C.$$

We conclude that

$$B(p-1) + \gamma + d^*_{[p,r]} > B(q-1) + \gamma + d^*_{[q,r]},$$

which contradicts the construction of $p = q_r$. Hence $\mathcal{P}$ is optimal.
Concerning time complexity we note: The procedure `FindBestPartition` consists of two nested loops with length $\leq n$. Operations within the loops are of $\mathrm{O}(1)$ yielding a time complexity of $\mathrm{O}(n^2)$. In the reconstruction part `SegmentationFromPartition` an evaluation of the functions $\mu_{[l+1,r]}$ is performed n times. So, if each evaluation is of order $\mathrm{O}(n)$ then Algorithm 1 works with a time complexity of $\mathrm{O}(n^2)$

For space consumption inspect the temporary variables used in the algorithm, all with sizes $\leq n+1$. Thus the algorithm has linear space complexity $\mathrm{O}(n)$. □

Let us finally indicate some modifications.

Remark 2 In applications we do not need to feed the algorithm with the exact values d_I^* of the functionals. There, the so-called m-estimates which are derived from one Newton step starting in the median of the data during the optimisation process related to M-estimation, could easily be plugged in. Since we can handle local medians efficiently (see below), this approach should prove powerful.

Instead of feeding the algorithm with the values d_I^* and μ_I^* stored in arrays one can also use fast evaluation schemes that also work in O(1) and have a memory consumption of O(n) only. Compare F. Friedrich (2005).

One may also restrict the search space.

Remark 3 Another potentially useful approach is to restrict the class of possible partitions. For example, we could require that the length of all intervals in the partition is greater than 2 (to avoid single spikes) or to be less than $\sqrt{n}$ (to avoid oversmoothing). The corresponding versions of the Bellman recursion (6) are obvious. Notice that the latter improves the time complexity to O($n^{3/2}$).

4 A Shooting Algorithm

One can compute minimising segmentations for *all parameters* γ *simultaneously* with moderate additional effort. This is due to the fact that one can easily determine a finite partition of the γ-axis into intervals, such that it is sufficient to compute optimal partitions of T on each of these intervals only once. To formulate this precisely we introduce some additional notation. First, recall that $\tilde{H}_\gamma$ and d_I^* do depend on input data $y \in \mathbb{R}^n$, which will be expressed by the index y in the following definitions. Let

$$G_y : [0,\infty) \longrightarrow \mathbb{R},\ \ \gamma \longmapsto \min_{\mathcal{P}\in\mathfrak{P}} \tilde{H}_\gamma(\mathcal{P}),$$

and set

$$\begin{aligned} b^k(y) &= \min\Big\{ \sum_{I\in\mathcal{P}} d_I^* : \mathcal{P}\in\mathfrak{P},\ |\mathcal{P}| = k\Big\}, \\ G_y^k &: [0,\infty) \longrightarrow \mathbb{R},\ \ \gamma \longmapsto \gamma(k-1) + b^k(y). \end{aligned}$$

The function G_y is the pointwise minimum of the finitely many affine functions G^k, i.e.

$$G_y(\gamma) = \min_{1\le k\le n} G_y^k(\gamma) \quad \text{for every } \gamma \ge 0. \tag{8}$$

The idea behind the following algorithm is to compute the intersection points γ_i of the affine functions G^k contributing to the minimum in (8) using Algorithm 1 for some values of the parameter γ. If we compute for some γ a minimising partition for $\tilde{H}_\gamma$ with cardinality k, this G^k is determined by the cardinality of the partition. For the two lines corresponding to the functions G^k and $G^{k'}$ extracted from neighbouring γ-values we compute the intersection point yielding a new candidate value for a γ_i. This is the reason for the word "shooting" in the title of this section. If the algorithms stops, we have computed for each $\gamma \ge 0$ partitions solving (5), see Corollary 2 below. Afterwards, minimising segmentations are obtained by

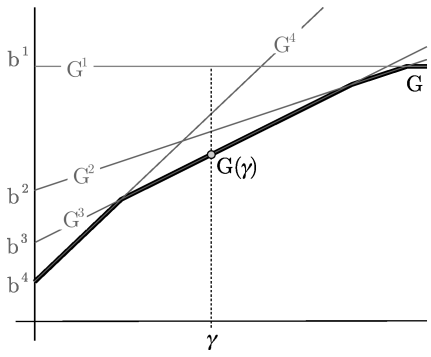

Figure 3: G is the pointwise minimum of affine functions G^k

adjoining the values μ_I^*. Let us formulate the results more general for a finite set of straight lines.

Let $n \in \mathbb{N}$ and $\Lambda \subset \mathbb{R}^2$ with $|\Lambda| = n$ be given. Consider straight lines with slope a and intercept b, $(a, b) \in \Lambda$. We are interested in the lowermost line function

$$F : \mathbb{R} \to \mathbb{R}, s \mapsto \min_{(a,b)\in\Lambda} s \cdot a + b.$$

Let $s \in \mathbb{R}$. In the sequel the symbols a_s^* and b_s^* will denote slope and intercept of a lowermost line at position s, i.e.

$$(a_s^*, b_s^*) \in \operatorname*{argmin}_{(a,b)\in\Lambda} s \cdot a + b.$$

Lemma 2 *The function*

$$F : \mathbb{R} \to \mathbb{R},\ s \mapsto s \cdot a_s^* + b_s^*$$

is concave, continuous and piecewise affine. Given lower and upper bounds $-\infty \le L < R \le \infty$, there are a natural number $1 \le m(\Lambda) \le n$ and parameters

$$L = \gamma_0 < \gamma_1 < \cdots < \gamma_{m(\Lambda)} = R, \tag{9}$$

such that (a_s^*, b_s^*) is well defined and constant for all $s \in (\gamma_{i-1}, \gamma_i)$ and such that $(a_s^*, b_s^*) \neq (a_t^*, b_t^*)$ if $s < \gamma_i < t$, $1 \le i \le m(\Lambda)$.

Let $s_i \in (\gamma_{i-1}, \gamma_i)$ for all $1 \le i \le m(\Lambda)$. Then $\gamma_i a_{s_i}^* + b_{s_i}^* = \gamma_i a_{s_{i-1}}^* + b_{s_{i-1}}^*$ and the mapping $i \mapsto a_{s_i}^*$ is strictly decreasing.

Remark 4 Note that on the intersection points γ_i the set $\operatorname{argmin}_{(a,b)\in\Lambda} \gamma_i \cdot a + b$ may consist of more than the two points $(a_{s_i}^*, b_{s_i}^*)$ and $(a_{s_{i-1}}^*, b_{s_{i-1}}^*)$.

Proof. Clearly, the point-wise minimum of finitely many continuous functions is a continuous function. The minimum of affine functions is concave:

$$\lambda \cdot \min_{(a,b)\in P} [s \cdot a + b] + (1-\lambda) \cdot \min_{(a,b)\in P} [t \cdot a + b] \le \min_{(a,b)\in P} [\lambda s + (1-\lambda)t + b].$$

Let $l, r \in \mathbb{R}$ with $l < r$. By definition of a^* and b^* the inequalities

$$r(a_r^* - a_l^*) \le b_l^* - b_r^* \le l(a_r^* - a_l^*) \tag{10}$$

hold and consequently $a_l^* = a_r^*$ implies $b_l^* = b_r^*$ and $a_l^* \neq a_r^*$ implies $a_l^* > a_r^*$. Therefore the set

$$\{s \in (L, R) : a_{s-\varepsilon}^* \neq a_{s+\varepsilon}^* \text{ for all } \varepsilon > 0\}$$

is finite. Thus F is piecewise affine with finitely many change points γ_i, $0 < i < m$ and the map $i \mapsto a_{s_i}$ is strictly decreasing. □

We will now apply the previous lemma to the functional G_y by considering $\Lambda = \{(k-1, b^k(y)) : 1 \leq k \leq n\}$. For each $k \in \mathbb{N}$ we denote the set of partitions with k pieces by $\mathfrak{P}_k = \{\mathcal{P} \in \mathfrak{P} : |\mathcal{P}| = k\}$ and the subset minimizing the data term by $\mathfrak{P}_k^* = \{\mathcal{P} \in \mathfrak{P}_k : \sum_{I\in\mathcal{P}} d_I^* \leq \sum_{I\in\mathcal{Q}} d_I^* \ \forall \mathcal{Q} \in \mathfrak{P}_k\}$. A direct consequence of the previous lemma reads as follows.

Corollary 1 *With notation from the previous lemma, let $\Lambda = \{(k-1, b^k(y)) : 1 \leq k \leq n\}$, $m(y) := m(\Lambda)$, $L = 0$, $R = \infty$ and $k(i) = a_{s_i}$ for all $1 \leq i \leq m(y)$. For the resulting parameters*

$$0 = \gamma_0 < \gamma_1 < \cdots < \gamma_{m(y)} = \infty, \tag{11}$$

the following holds:

(i) For each $i = 1, \ldots, m(y)$ and each $\gamma \in (\gamma_{i-1}, \gamma_i)$ we have $\operatorname{argmin} \tilde{H}_\gamma = \mathfrak{P}_{k(i)}^*$.

(ii) For each γ_i, $i = 0, \ldots, m(y) - 1$, the set $\operatorname{argmin} \tilde{H}_{\gamma_i}$ is the union of those $\mathfrak{P}_k^*$ for which $G_y(\gamma_i) = G_y^k(\gamma_i)$ and contains both $\mathfrak{P}_{k(i)}^*$ and, if $i \geq 1$, $\mathfrak{P}_{k(i-1)}^*$.

(iii) For each $\gamma \in (\gamma_{m(y)-1}, \infty)$ the functional $\tilde{H}_\gamma$ has $\{T\}$ as unique location of the minimum.

Proof. To see (i) and (ii), replace the function F in Lemma 2 by G displayed in (8). To see (iii), observe that $\lim_{\gamma\to\infty} G^k(\gamma) = \infty$ for all $k \geq 1$ and therefore obtain $G(\gamma) = G^0(\gamma)$ for large γ. Correspondingly, only partitions $\mathcal{P}$ with $|\mathcal{P}| = 1$ can minimise (1). Since $|\mathcal{P}| = 1$ is equivalent to $\mathcal{P} = T$ this proves (iii). □

The following lemma provides the keys for the development of a recursive algorithm to determine the gamma scale (9).

Lemma 3 *Let $l, r \in \mathbb{R}$ with $l < r$ and $a_l^* \neq a_r^*$. Then there is an intersection point q with $l \leq q \leq r$ and*

$$a_l^* \cdot q + b_l^* = a_r^* \cdot q + b_r^*. \tag{12}$$

Additionally, one and only one of the following two cases occurs:

(1) $q \cdot a_q^* + b_q^* = q \cdot a_l^* + b_l^* = q \cdot a_r^* + b_r^*$,

(2) $q \cdot a_q^* + b_q^* < q \cdot a_l^* + b_l^* = q \cdot a_r^* + b_r^*$.

(1) implies $F(s) = s \cdot a_l^* + b_l^*$ for all $s \in (l, q]$ and $F(s) = s \cdot a_r^* + b_r^*$ for all $s \in [q, r)$.
(2) implies the inequalities $l < q < r$ and $a_l^* > a_q^* > a_r^*$.

Proof. In the proof of Lemma 2 we found the inequalities $r(a_r^* - a_l^*) \leq b_l^* - b_r^* \leq l(a_r^* - a_l^*)$. Since by the same lemma a^* is decreasing and (12) means $(b_r^* - b_l^*) = q(a_r^* - a_l^*)$ we obtain $l \leq q \leq r$. By definition $q \cdot a_q^* + b_q^* \leq q \cdot a_s^* + b_s^*$ for all $s \in \mathbb{R}$ and therefore only the two cases (1) and (2) can occur.

If (1) holds then a convex combination of the inequalities $q \cdot a_l^* + b_l^* = q \cdot a_q^* + b_q^* \leq q \cdot a_s^* + b_s^*$ and $l \cdot a_l^* + b_l^* \leq l \cdot a_s^* + b_s^*$ for all $s \in \mathbb{R}$ implies that $s \cdot a_l^* + b_l^* \leq s \cdot a_s^* + b_s^*$ for all $l \leq s \leq q$. The same result holds for (a_r^*, b_r^*) and the statement about (1) is proved.

The inequality $l < q < r$ follows immediately from the inequality in (2) and $a_l^* > a_q^* > a_r^*$ is a consequence of $r(a_r^* - a_q^*) \leq b_q^* - b_r^* \leq l(a_r^* - a_q^*)$ and the corresponding inequality for a_l^* and a_q^*. □

The following is a recursive procedure used for the computation of the list (9).

Procedure `BuildGammaList`($a_l, b_l, a_r, b_r \in \mathbb{R}$; *List* `lst`)

output: list of values $\gamma \in \mathbb{R}$ appended to `lst`
local : intersection point $q \in \mathbb{R}$; line parameters $a_q, b_q \in \mathbb{R}$

begin
  $q \leftarrow \frac{b_r - b_l}{a_l - a_r}$ (* intersection point of G^l and G^r *);
  $(a_q, b_q) \leftarrow$ `GetLine`(q);
  if $q \cdot a_q + b_q = q \cdot a_l + b_l$ **then**
    `Append`(`lst`, q);
  else
    `BuildGammaList`(a_l,b_l,a_q,b_q,`lst`); `BuildGammaList`(a_q,b_q,a_r,b_r,`lst`);
  end
end

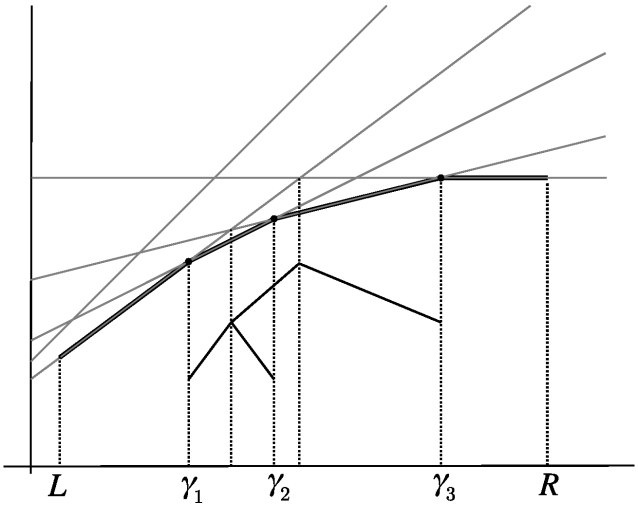

Figure 4: γ-shooting: the displayed tree represents the data structure implicitly created when `BuildGammaList` is called recursively.

Algorithm 2: γ-Shooting

input : left and right interval bounds $L < R$. Procedure `GetLine` to determine lowermost line.
output: list `lst` containing values $\gamma \in \mathbb{R}$
local : line parameters (a_L, b_L) and (a_R, b_R)

```
begin
    --- initialisation for γ = 0 ---
    (a_L, b_L) ← GetLine(L); (a_R, b_R) ← GetLine(R);
    --- build list ---
    lst := CreateEmptyList;
    if a_L ≠ a_R then
     | BuildGammaList(a_L,b_L,a_R,b_R,lst);
    end
    if L ∉ lst then
     | InsertBefore(L,lst);
    end
    if R ∉ lst then
     | Append(lst, R);
    end
end
```

Theorem 3 *Let $L, R \in \mathbb{R}$ with $L < R$ and $M = |\{1 \le i \le m(\Lambda) : L \le \gamma_i \le R\}|$. Then there is a constant $c > 0$ such that, if there is a number $N \in \mathbb{N}$ and a procedure* `GetLine` *that for each $q \in \mathbb{R}$ returns the slope a_q^* and intercept b_q^* within at most N steps, Algorithm 2 determines the list (9) in $c \cdot N \cdot M$ steps.*

Proof. Let $l < r$. By Lemma 3 only one of the following cases can occur, if procedure `BuildGammaList` is called with parameters (a_l^*, b_l^*) and (a_r^*, b_r^*):
(1) The value q is inserted into the list and in the intervals (l, q) and (q, r) the graph of G is uniquely determined by the two lowermost lines (a_l^*, b_l^*) and (a_r^*, b_r^*). If l and r are not lowermost intersection points, then consequently $q \neq l$ and $q \neq r$.
(2) `BuildGammaList` is again called with $a_l^* > a_q^* > a_r^*$, where q is no lowermost intersection point.
Since only the cases (1) and (2) can occur, the procedure `BuildGammaList` either terminates (and enters a value in the list) or it recurses with values a_l^*, a_q^* and a_q^*, a_r^* corresponding to intervals strictly contained in $[a_l^*, a_r^*]$. By the principle of nested intervals the procedure thus terminates and determines the list of values as described in Lemma 2. The statement about complexity is also derived from the principle of nested intervals since `BuildGammaList` is called with $a_l^* > a_q^* > a_r^*$. If L or R are lowermost intersection points, then they may or may not be inserted in the list. This is checked by the two `if`-statements. □

Now we come back to the Potts functional. The following procedure can be used for the determination of line parameters associated to data y or more generally, to distances d^*. We assume that the variables d_I^* are accessible as global variables for each interval $I \subset \{0, \dots, n\}$.

Procedure `GetPottsLine`$(\gamma \in \mathbb{R}) \in \mathbb{R}^2$

output: line parameters $(a, b) \in \mathbb{R}^2$
global : data length $n \in \mathbb{N}$; distances $d^* \in \mathbb{R}^n \times \mathbb{R}^n$
local : partition vector $p \in \mathbb{N}^n$, left and right interval bounds: $l, r \in \mathbb{N}$; temporary variable $t \in \mathbb{N}$

begin
  if $\gamma = \infty$ **then**
    $a \leftarrow 1$; $b \leftarrow d^*_{[1,n]}$
  else
    $p \leftarrow$ `FindBestPartition`(n,γ,d^*);
    `--- compute cardinality and distance of partition ---`
    $r \leftarrow n$; $l \leftarrow p_r$; $(a, b) \leftarrow (0, 0)$;
    while $r > 0$ **do**
      $a \leftarrow a + 1$; $b \leftarrow b + d^*_{[l+1,r]}$;
      $r \leftarrow l$; $l \leftarrow p_r$;
    end
  end
  return (a, b);
end

Procedure `GetPottsLine` allows us to determine elements from the set of lines $\Lambda(y)$ described at the beginning of this paragraph. Algorithm 2 combined with this procedure can be used to determine the $m(y) := m(\Lambda(y))$ lowermost intersection points γ_i, $1 \le i \le m(y)$:

Corollary 2 *Algorithm 2 used with $L = 0$, $R = \infty$ and procedure* `GetLine=GetPottsLine` *terminates and computes the intersection points γ_i in (11). If evaluations of the functions $d^*_{[l,r]}$ $(1 \le l \le r \le n)$ at time points $l \le x \le r$ take $\mathrm{O}(1)$ steps the algorithm works with a time complexity $\mathrm{O}(n^2 \cdot m(y))$ and space complexity $\mathrm{O}(n \cdot m(y))$.*

Proof. The statements are direct consequences of Theorem 2 and Theorem 3. □

Remark 5 In applications, the if statement checking for $\gamma = \infty$ in procedure `GetPottsLine` can be replaced by feeding Algorithm 2 with a very large value for R.

The algorithm is an improvement of one suggested early by some of the authors.

Remark 6 In V. Liebscher and G. Winkler (1999) we reported on a $\mathrm{O}(n^3)$ time, $\mathrm{O}(n^2)$ space, algorithm based solely on dynamic programming. The present algorithm is better in both aspects. This is clear for the space consumption. As far as time complexity is concerned, numerous numerical simulations (both on real data and test beds) suggest that generically $m(y)$ is considerably smaller than n. Moreover, consider typical time series which have sound representation with few jumps. Then other segmentations can only survive if γ is very small. In fact, in simulations the overwhelming number of γ-intervals appear close to $\gamma = 0$.

Skipping these intervals leads to $m(y) \ll n$ and thus a further significant improvement over $\mathrm{O}(n^3)$.

5 ℓ^p-Loglikelihoods

By Corollary 2, the crucial point is to compute all $d^*_{[l,r]}$, $1 \le l \le r \le n$, efficiently. So, let us address optimisation of the distances d_I now, in order to illustrate the scope and thereby underline the relevance of the algorithms. We will restrict ourselves to distances d_I of the form

$$d_I(y_I, \mu_I) = \sum_{i \in I} \varrho(y_i - \mu_I(i)),$$

with a real function $\varrho(u)$ on $\mathbb{R}$. Clearly, all (negative) loglikelihood functions of location families are of this type. Minima exist under natural conditions, for example that ϱ be symmetric around 0, increasing in $|u|$, and lower semicontinuous.

The most important examples of ϱ are the ℓ^p-norms, in particular the common case $p = 2$, and $p = 1$. The latter received increasing interest recently. Other examples are the convex 'Huber functions', which are quadratic in a symmetric neighbourhood around zero, and linearly increasing outside, or the non-convex ones of the 'Hampel type', like truncated squares, or all other functions appearing in the context of M-estimation of location, cf. F. Hampel et al. (1986).

To fix the ideas, let us consider the most common example. Suppose we want to minimise

$$\sum_{i \in I} \varrho(y_i - p(t_i)), \quad p(t_i) = \sum_{j=0}^{l} c_j \phi_j(t_i), \ \ (c_j)_{j=1}^{l} \in \mathbb{R}^l, \tag{13}$$

on an interval $I \subset T$ for a system of basis functions $\phi_1, \ldots, \phi_l$ and with distinct design points $t_i \in \mathbb{R}$, $i \in I$. Then $\mathcal{F}_I$ becomes the linear space of the vectors $(\mu_I(i) = p(t_i) : i \in I)$ and, to compute μ_I^* and d_I^*, optimal coefficients c_j^* must be determined.

Efficient access to relevant quantities can be based on the following simple observation.

Lemma 4 *For every function $\psi : T \mapsto \mathbb{R}$ there is a $\mathrm{O}(n)$-tabulation from which each of the values $\sum_{i \in I} \psi(i)$ for intervals I in T can be computed in $\mathrm{O}(1)$.*

Proof. The tabulation of the n values $\sum_{i=1}^{k} \psi(i)$, $k = 1, \ldots, n$ has time complexity $\mathrm{O}(n)$. For each $I = [a, b]$ the values

$$\sum_{i \in I} \psi(i) = \sum_{i=1}^{b} \psi(i) - \sum_{i=1}^{a-1} \psi(i)$$

are computed in $\mathrm{O}(1)$ which completes the proof. □

The ℓ^2-case $\varrho(u) = u^2$ is ubiquitous in the literature on regression and approximation theory. One simply has to solve the normal equations. In the simplest case, where $\mathcal{F}_I$ consists of the constant time series, one gets sums

$$d_I(y_I, \mu_I) = \sum_{i \in I} (y_i - \mu_I)^2 \tag{14}$$

of squared deviations from data, and concerning minimisation,

$$\min_{\mu_I \in \mathbb{R}^I} \sum_{i \in I} d_I(y_I, \mu_I) = \min_{a_I \in \mathbb{R}} \sum_{i \in I} (y_i - a_I)^2 = \sum_{i \in I} (y_i - \bar{y}_I)^2.$$

Let us come back to the general ℓ^2-case and continue from Lemma 4. The number $m(y)$ was defined in Corollary 1.

Proposition 1 *Let a system of basis functions $\phi_1, \ldots, \phi_l$, and mutually distinct design points $t_i \in \mathbb{R}$, $i \in T$, be given, and consider the regression problems from (13) with*

$$d_I(y_I, \mu_I) = \sum_{i \in I} (y_i - p(t_i))^2. \tag{15}$$

Then a location of the minimum for the functional (3) can be computed in time complexity $\mathrm{O}(n^2)$ for a single $\gamma \geq 0$, and in time complexity $\mathrm{O}(m(y)n^2)$ for all γ simultaneously, space complexity is $\mathrm{O}(n)$ in either case.

Proof. For each interval $I \subset T$ we have to solve the normal equations

$$\sum_{i \in I} y_i \phi_k(t_i) = \sum_{j=1}^{l} c_j \sum_{i \in I} \phi_j(t_i)\phi_k(t_i), \quad k = 1, \ldots, l.$$

We may resort to any standard numerical method. Given the cumulative moments on both sides, the algorithms work in time complexity $\mathrm{O}(l^3)$ which does not depend on the sizes $|I|$ of intervals. By Lemma 4, the cumulative moments can be computed in $\mathrm{O}(n)$. The rest follows immediately from Theorem 2 and Corollary 2. □

Recall that the regression spaces $\mathcal{F}_I$ may vary from interval to interval.

Example 3 Consider regression spaces $\mathcal{F}_I$ of polynomials ϕ_j^I with length-dependent maximal degrees $r(|I|)$. After tabulations in time $\mathrm{O}(n^2)$, similar to those sketched above, solutions can be computed in time complexity $\mathrm{O}(\sum_{k=1}^{n}(n-k)r(k)^3)$. Since it does not make sense to use polynomials of degree higher than $n-2$, time complexity is at most $\mathrm{O}(n^5)$ both for a single γ, and for all $\gamma > 0$ simultaneously. If we decide on polynomials of degree smaller than a fraction of n, say $r(n) = n^{1/2}$, time complexity becomes $\mathrm{O}(n^{7/2})$.
Similar arguments apply to unisolvable systems of general basis functions. For an account we refer to L. Györfi et al. (2002).

We discuss now two examples which are related to piecewise constant ℓ^2-regression.

Example 4 Up to now, we considered least squares where the scale of noise is assumed to be known. Among others, Y.-C. Yao (1988) and Y.-C. Yao and S. Au (1989) studied estimation of piecewise constant functions degraded by Gaussian white noise with *unknown* variance. This case is very important in practice.
Schwarz' model choice criterion recommends to minimise

$$\gamma \cdot |J(x)| - \frac{1}{2} \ln \sum_{i=1}^{n} (y_i - x_i)^2 \tag{16}$$

for $\gamma = \ln n/2$. For Akaike's AIC-criterion the constant is $\gamma = 1$.
Because of the nonlinear logarithm, the usual Bellman recursion does not work any more. Extending the recursion by including the reconstruction complexity as a second parameter, similar to V. Liebscher and G. Winkler (1999), with a new Bellman function

$$B(k,l) = \inf_{(\mathcal{P},\mu) \in \mathfrak{S}(k), \#\mathcal{P} = l} \Big(\gamma(|\mathcal{P}| - 1) + \sum_{I \in \mathcal{P}} d_I^* \Big), \;\; k \geq 1, 1 \leq l \leq k-1,$$

a double recursion over k and l solves the problem in $\mathrm{O}(n^3)$ time for all γ simultaneously.
In contrast to this, the heteroscedastic version of the functional (16) can be minimised within the main scheme of the present paper. Then, for each interval there are two parameters μ_I and σ_I^2. The Gaussian negative loglikelihood for an interval I reads

$$d_I(\mu_I, \sigma_I^2, y) = \frac{1}{2\sigma_I^2} \sum_{i \in I} (y_i - \mu_I)^2 + \frac{1}{2} \ln \sigma_I^2.$$

Standard optimisation yields $\mu_I^* = \bar{y}_I$ and $(\sigma_I^2)^* = (1/|I|) \sum_{i \in I} (y_i - \bar{y}_I)^2$ and therefore

$$d_I^* = \frac{1}{2} + \frac{1}{2} \ln \sum_{i \in I} (y_i - \bar{y}_I)^2.$$

If $|I| = 1$ one has $(\sigma_I^2)^* = 0$ and there is no loglikelihood. Therefore, we have to restrict this procedure to intervals with at least two points, see also Remark 3.
Obviously, computation has the same time and space complexity as in the standard ℓ^2-case in Proposition 1.

Let us now turn to the more robust ℓ^1-case and, again, start with spaces $\mathcal{F}_I$ of constant functions. Then (14) is replaced by the sums

$$d_I(y_I, \mu_I) = \sum_{i \in I} |y_i - \mu_I| \tag{17}$$

of total deviations from data. For each interval I, the minimum is attained in the median of data $(y_i)_{i \in I}$. As a *pendant* to Proposition 1 we get

Proposition 2 *Let each $\mathcal{F}_I$ consist of all constant functions, and be equipped with the distance d_I in (17). Then a location of the minimum for the functional (2) can be computed in time complexity $\mathrm{O}(n^2 \ln n)$ for a single $\gamma \geq 0$, and in time complexity $\mathrm{O}(n^3)$ for all γ simultaneously. In either case, space complexity is $\mathrm{O}(n^2)$.*

Proof. First we have to determine the local medians. For each single $i \in T$ and increasing $j \geq i$, we sort data y_k, $i \leq k \leq j$ in increasing order. Then their median over $I = [i, j]$ can be determined in constant time. Furthermore, using so-called red-black trees from T. Cormen et al. (2001), Chapter 15, we can sort data $y_{[i,j+1]}$ in $\mathrm{O}(\ln n)$ time if data $y_{[i,j]}$ are already sorted. Then the median can be retrieved in $\mathrm{O}(\ln n)$ time too. Hence the medians of data over all intervals in T can be computed in $\mathrm{O}(n^2 \ln n)$ time.
By Lemma 4, for each $j \in T$ and each interval I, the values $\sum_{i \in I} |y_i - y_j|$ can be computed in constant time after a tabulation which needs quadratic time. Afterwards, for each interval I, one computes the value d_I^* in constant time. To complete the procedure, Algorithms 1, or 2, respectively, are applied.
Space consumption of both procedures is quadratic and the space consumption for tabulation is quadratic too. This completes the proof. □

We conclude the discussion of the ℓ^1-case with a final remark.

Remark 7 Concerning regression, the ℓ^1-case is much more intricate and unpleasant than the ℓ^2-case. Nevertheless, the ℓ^1-theory has historically older roots than the ℓ^2-theory; recent interest in this case is mainly due to questions about robustness. We refer to

P. Bloomfield and W. Steiger (1983) for a preliminary account. Complexity is usually connected to that of the simplex algorithm; the cited authors report a rate of $O(|I| \ln |I|)$ instead of the rate $O(|I|)$ in the above ℓ^2-case.

Let us as a third and last case briefly mention local absolute deviation, or the ℓ^∞-approach, again for constant regression. We insert for (14) or (17) the expression

$$d_I(y_I, \mu_I) = \max_{i \in I} |y_i - \mu_I|.$$

The minimum is then attained by the midrange

$$d_I^* = (\max_{i \in I}\{y_i : i \in I\} - \min_{i \in I}\{y_i : i \in I\})/2.$$

Again, we obtain the same complexity results as in the ℓ^2 case since (6) is true.

We finally comment briefly on the global ℓ^∞-case, for which this recursion is not valid any more. The functional is determined by

$$D((\mathcal{P}, \mu), y) = \max_{I \in \mathcal{P}} \max_{i \in I} |y_i - \mu_I|.$$

Furthermore, for getting minimal points of this functional, μ_I^* need not be a minimal point of d_I. But, the midrange of data y_i, $i \in I$, is again one valid choice for a minimiser and instead of (7) we find

$$B(k) = \min\big(\min_{1 \le j \le k-1} \gamma + \max(B(j), d^*_{[j+1,k]}),\big), \;\; k \ge 1. \tag{18}$$

The main difference to (7) is the substition of '+' by 'max'. Hence we obtain the same complexities of the recursion parts as above. Tabulation can be done in $O(n^2)$ space and time complexity.

Let us finally sketch an application in molecular biology.

Example 5 We applied the algorithms to fractionation experiments for cDNA-microarrays established in A. Drobyshev et al. (2003). For each spot on a chip a time series of length 29, called fractionation curve, is recorded. The most informative features are abrupt intensity changes, which characterize if there is (undesired) cross-hybridization or not. Hence they are valuable indicators of the quality of the single spots. Fig. 5 displays three different types of fractionation curves, indicating from left to right zero, one, and two jumps down. With spaces $\mathcal{F}_I$ of constant functions, the output of the above algorithm for the ℓ^2-likelihood is contrasted to that for the ℓ^1-likelihood variant. The change points identified by the two methods are similar, especially for large γ-values (upper plots). On the other hand, there is ample evidence that the height of jumps returned by the ℓ^1-algorithm is a much more reliable estimate than that returned by the ℓ^2-algorithm.

To illustrate the power of the ideas behind the algorithms, let us conclude with an example which goes beyond the scope of ℓ^p-loglikelihood scores. It is concerned with counting data governed by generalised linear models.

Example 6 We consider piecewise constant regression under a binomial model. Let $(y_i)_{i=1}^n$ be a sample from independent random variables with binomial distributions of size $n_i \in \mathbb{N}$ and parameter $\mu_I \in [0, 1]$ if $i \in I$. The sample sizes n_i are supposed to be known, and the interval probabilities μ_I have to be estimated.

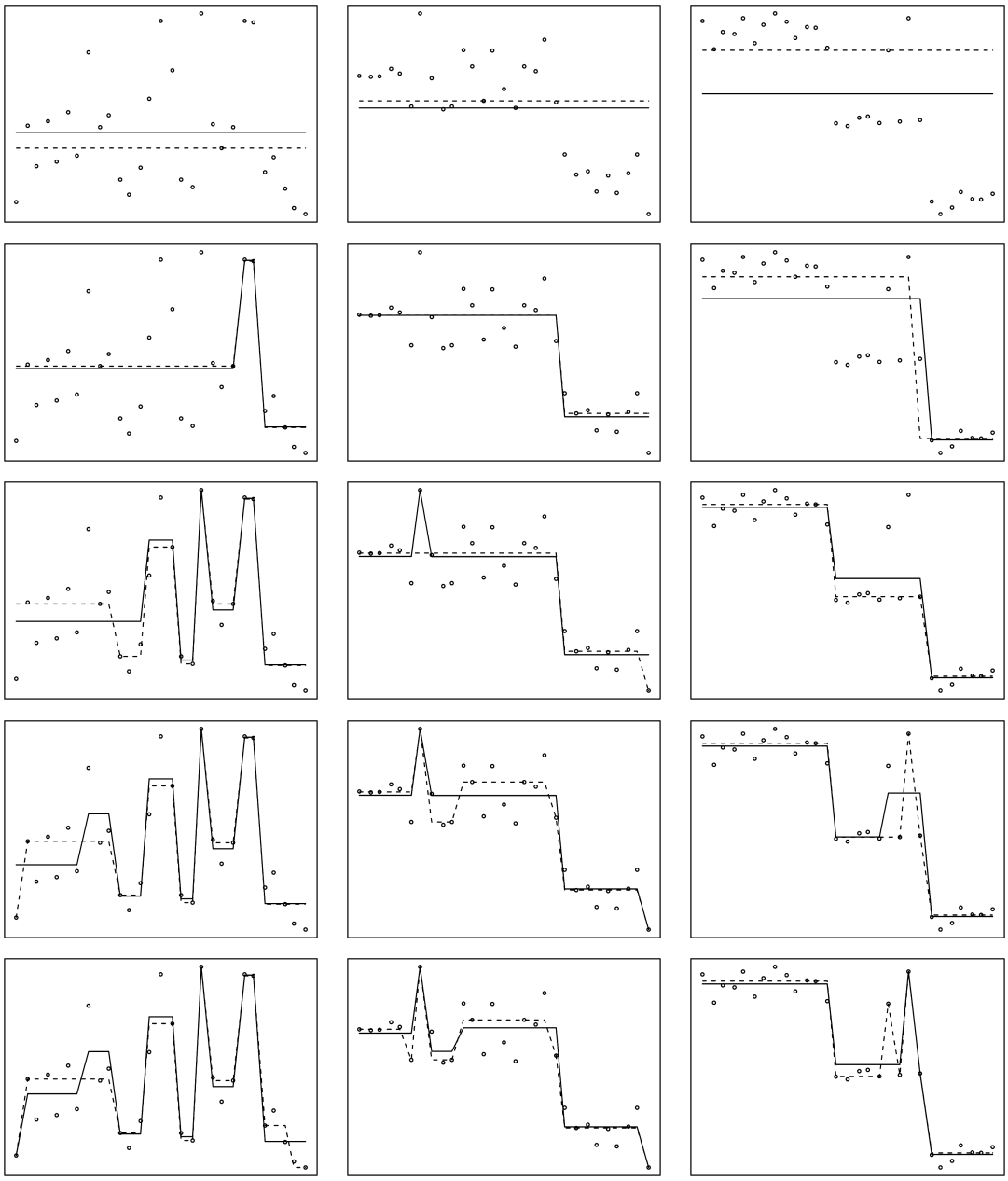

Figure 5: Gene expression data: three different types of fractionation curves, each coloumn one, data displayed as dots. Output of the ℓ^1-algorithm (dashed lines) contrasted to the ℓ^2-algorithm (solid lines), for the respective five rightmost γ-intervals in decreasing order.

The log-likelihood score is

$$d_I(y_I, \mu_I) = \sum_{i \in I} y_i \ln \mu_I + \sum_{i \in I} (n_i - y_i) \ln(1 - \mu_I).$$

The maximum likelihood estimate for each interval I from a given partition is

$$\mu_I^* = \sum_{i \in I} y_i \Big/ \sum_{i \in I} n_i,$$

and the corresponding optimal values d_I^* can be computed. Again both, Algorithm 1 and 2, apply to the computation of the complexity penalised maximum likelihood estimates. This yields, due to the efficient computation of local means in Lemma 4, time complexities $\mathrm{O}(n^2)$ and $\mathrm{O}(n^3)$ respectively.

6 Weak and Morphological Constraints

Functionals of seemingly completely different flavour than that of the introductory example are covered by the framework marked out in Section 2. In this final section, we present a selection of typical examples.

The first one addresses local weak smoothness constraints. A nowadays classical instance is the *Blake-Zissermann functional*. It was proposed in the early 1980^{th} in A. Blake (1983) and A. Blake and A. Zisserman (1987), and independently in S. Geman and D. Geman (1984) for discrete intensity values. For time series, the original version has the form

$$BZ_{\gamma,\tau}(x, y) = \sum_{i=1}^{n} \min\{\tau^2 (x_{i+1} - x_i)^2, \gamma)\} + \sum_{i=1}^{n} (y_i - x_i)^2. \tag{19}$$

The function $\min\{(\tau u)^2, \gamma\}$ in the first term is a truncated square function with width $2\gamma^{1/2}/\tau$ and height γ of the 'cup'. It appears in (robust) M-estimation where it is introduced for example in D. F. Andrews et al. (1972). Clearly, the functional (1) is the degenerate case of (19) for $\tau \to \infty$.

An equivalent formulation of the associated minimisation problem in the formalism from (2) and (3) reads as follows: Let $\mathcal{F}_I = \mathbb{R}^I$ and define a function of segmentations by

$$\widetilde{BZ}_{\gamma,\tau}(\mathcal{P}, \mu) = \gamma(|\mathcal{P}| - 1) + \sum_{I \in \mathcal{P}} \sum_{i, i+1 \in I} \tau^2 (\mu(i+1) - \mu(i))^2 + \sum_{i \in T} (y_i - \mu(i))^2. \tag{20}$$

This means that, given a partition, there are local sums of squared deviations of neighbouring intensities inside the intervals of the partition, and a penalty γ for each break between adjacent intervals. Taking minima for both functionals reveals that x^* is a location of the minimum for (19) if and only if $(\mathcal{P}^*, x^*)$ is one for (20) with $\mathcal{P}^*$ defined by the time points $i \in T$ where $|x_{i+1}^* - x_i^*| \leq \gamma^{1/2}/\tau$. This is shown and discussed in connection with robustness in G. Winkler and V. Liebscher (2002) and G. Winkler et al. (1999).

The functional (20) is of the form (2) with

$$d_I(y_I, \mu_I) = \langle \mu_I, (\tau^2 B + \mathrm{Id}) \mu_I \rangle - 2\langle \mu_I, y_I \rangle + \langle y_I, y_I \rangle, \tag{21}$$

where $\langle \cdot\,, \cdot \rangle$ is the Euclidean inner product on $\mathbb{R}^I$, and A is the $|I| \times |I|$-bandmatrix

$$B = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & 0 & \\ 0 & & \ddots & & 0 \\ & 0 & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}.$$

The quadratic minimisation problem for (21) has the unique shrinkage-type solution

$$\mu_I^* = (\tau^2 B + \mathrm{Id})^{-1} y_I.$$

For the computation of $(\tau^2 B + \mathrm{Id})^{-1}$ one can use the eigenvalues and -vectors of A. One derives eigenvalues $\lambda_k = 2(1 - \cos((k-1)\pi/n)$ and the k^{th} eigenprojection Pr_k onto their eigenspaces spanned by the eigenvectors $u_1 \equiv 1$, and u_k, $2 \le k \le n$, with components

$$\sin((k-1)\pi/n \cdot i) - \sin((k-1)\pi/n \cdot (i-1)), \quad 1 \le i \le n,$$

see for example H. Künsch (1994); T. S. Chihara (1978) is a standard reference for the background.

In contrast to the above scheme, $d_I(y_I, \mu_I^*)$ now is nonlinear in τ^2. In fact, we have

$$d_I^* = d_I(y_I, \mu_I^*) = \langle y_I, (\mathrm{Id} - (\tau^2 B + \mathrm{Id})^{-1}) y_I \rangle = \sum_{k=1}^{|I|} \frac{\tau^2 \lambda_k}{\tau^2 \lambda_k + 1} \langle y_I, \mathrm{Pr}_k\, y_I \rangle.$$

Due to this nonlinearity in τ^2, it is somewhat harder to implement the scanning of minimisers of the functional (20) as a function of both parameters τ^2 and γ. Nevertheless, for constant τ we can employ our algorithms. All what is necessary is to compute the local moments of y with the functions $\sin(k\pi \cdot /n)$ and $\cos(k\pi \cdot /n)$ for $k = 1, \ldots, n$.

This can be done in quadratic tabulation time and the same computational complexities (quadratic respectively cubic) as computed above. We conclude, that for one dimensional time this algorithm is a fast, exact, and convenient alternative to the graduated nonconvexity algorithm (GNC) proposed by A. Blake and A. Zisserman (1987), not to speak about simulated annealing.

A completely different way to impose weak constraints is to restrict the function spaces $\mathcal{F}_I$. In nonparametric statistics, in particular if there is little groundtruth, qualitative features are of special interest. Let us give two examples of morphological features, we found especially suited for specific data sets, for example from gene expression and brain mapping. For a thorough discussion cf. G. Winkler et al. (2005).

For the first example, let $\mathcal{F}_I$ be the set of those time series which either increase or decrease on I. We will refer to this case as *piecewise monotonic regression*.

Theorem 4 *The complexity penalised piecewise monotonic regression problem of minimising the functional (15) can be solved in time complexity* $\mathrm{O}(n^2)$ *for a single* $\gamma \ge 0$*, and in time complexity* $\mathrm{O}(n^3)$ *for all* γ *simultaneously. In both cases, space complexity is* $\mathrm{O}(n^2)$*.*

Proof. We have to compute both, increasing and decreasing regressions for each interval. The corresponding pool adjacent violators algorithm (PAVA, see M. Ayer et al. (1955)) computes this regression in linear time. We start this algorithm at each point $i \in T$

separately. Then in the k^{th} step of PAVA, the increasing and decreasing regressions for the interval $[i, i+k]$, are already computed. Consequently, the increasing and decreasing regressions for all intervals $[i, j]$ are computed in $\mathrm{O}(n^2)$ time by this scheme. In view of Theorem 2 and Corollary 2 this completes the proof. □

We conclude the discussion of monotone regression by a remark concerning the ℓ^1-case.

Remark 8 For ℓ^1-monotonic regression there are algorithms similar to the pool adjacent violators algorithm with complexity $\mathrm{O}(n \ln n)$, see V. Boyarshinov and M. Magdon-Ismail (2004). This way, algorithms with similar complexities but with some additional logarithmic factors can be derived.

There are even more sophisticated examples for morphological penalties.

Example 7 Let us consider mode penalised least squares regression with a penalty counting the number of modes in a signal x instead of its jumps. In P. L. Davies (1995) and also in P. L. Davies and A. Kovac (2001) the (low) number of modes is a crucial quality measure for the parsimonious explanation of time series data.
We associate to each partition those signals, which increase and decrease on each pair of subsequent intervals, or conversely. This restriction requires a modification of the Bellman equations, which we are going to sketch now. First, we introduce two Bellman functions $B_\pm$ given by

$$B_\pm(k) = \min_{(\mathcal{P},\mu)\in\mathfrak{S}(k),\mu\in\mathcal{F}_\mathcal{P}^\pm} \Big(\gamma(|\mathcal{P}|-1) + \sum_{I\in\mathcal{P}} d_I^*\Big),\ k \geq 1.$$

The symbol $\mathcal{F}_\mathcal{P}^+$ denotes the space of all signals in $[1,k]$, which are increasing in the last interval of $\mathcal{P}$ (under the natural order) and similarly $\mathcal{F}_\mathcal{P}^-$ the space of all signals in $[1,k]$, which are decreasing there. Denoting by $d_I^{*,\pm}$ the minimal values for increasing and decreasing regression, we obtain recursion formulae of the form

$$B_\pm(k) = \min_{0\leq j\leq k-1} B_\mp(j) + \gamma + d^{*,\pm}_{[j+1,k]},\ k \geq 1.$$

Since every minimal point of the functional (15) has to realise either $B_+(n)$ or $B_-(n)$, these recursions allow to solve mode penalised least squares regression with the same complexities as in Theorem 4.

Summarising the above derivations, let us emphasize once more that the dynamic programming approach is able to solve a lot of important optimisation problems from complexity penalised M-estimation without need to specify the hyperparameter in advance. This will prove useful in applications, like indicated in Example 5. Simulations and tests were performed with the software package AntsInFields; a CD-ROM is attached to G. Winkler (2003), free download under F. Friedrich (2003). A partial implementation of the algorithms is contained in the R package tssegmentation, which can be obtained from V. Liebscher (2005).

References

[1] D. F. Andrews, P. J. Bickel, F. R. Hampel, P. J. Huber, W. H. Rogers, and J. W. Tukey. *Robust estimates of location. Survey and advances.* Princeton University Press, Princeton, N. J., 1972.

[2] M. Ayer, H. Brunk, G. Ewing, W. Reid, and E. Silverman. An empirical distribution function for sampling with incomplete information. *Ann. Math. Stat.*, 26:641–647, 1955.

[3] P. Bhattacharya. Some aspects of change-point analysis. In E. C. et al., editor, *Change-point problems. Papers from the AMS-IMS-SIAM Summer Research Conference held at Mt. Holyoke College, South Hadley, MA, USA, July 11-16, 1992.*, volume 23 of *IMS Lect. Notes, Monogr. Ser.*, pages 28–56 , Hayward CA, 1994. Institute of Mathematical Statistics.

[4] A. Blake. The least disturbance principle and weak constraints. *Pattern Recognition Lett.*, 1:393–399, 1983.

[5] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press Series in Artificial Intelligence. MIT Press, Massachusetts, USA, 1987.

[6] P. Bloomfield and W. Steiger. *Least Absolute Deviations. Theory, Applications, and Algorithms*, volume 6 of *Progress in Probability and Statistics*. Birkhäuser, Boston, Basel, Stuttgart, 1983.

[7] V. Boyarshinov and M. Magdon-Ismail. Linear time isotonic and unimodal regression in the L_1 and L_∞. Technical Report TR 04-02, RPI Computer Science, January 2004.

[8] L. Boysen, V. Liebscher, A. Munk, and O. Wittich. Jump-penalized least squares: Consistencies and rates of convergence. Schriftenreihe des IBB, 05-3, January 2005. submitted.

[9] T. S. Chihara. *An introduction to orthogonal polynomials*, volume 13 of *Mathematics and its Applications*. Gordon and Breach Science Publishers, New York, London, Paris, 1978.

[10] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, 2nd edition, 2001.

[11] M. Csörgö and L. Horváth. *Limit theorems in change-point analysis*. John Wiley & Sons., Chichester, 1997.

[12] P. L. Davies. Data features. *J. of the Netherlands Society for Statistics and Operations Research*, 49(2):185–245, July 1995.

[13] P. L. Davies and A. Kovac. Local extremes, runs, strings and multiresolution. *Ann. Stat.*, 29(1):1–65, 2001.

[14] D. Donoho. Wedgelets: Nearly minimax estimation of edges. *The Annals of Statistics*, 27(3):859–897, 1999.

[15] A. Drobyshev, C. Machka, M. Horsch, M. Seltmann, V. Liebscher, M. Hrabè de Angelis, and J. Beckers. Specificity assessment from fractionation experiments (SAFE): A novel method to evaluate microarray probe specificity based on hybridization stringencies. *Nucleic Acids Res.*, 31(2):1–10, 2003.

[16] F. Friedrich. AntsInFields: Stochastic simulation and Bayesian inference for Gibbs fields, 2003. URL `www.AntsInFields.de`.

[17] F. Friedrich. *Complexity Penalized Segmentations in 2D - Efficient Algorithms and Approximation Properties*. PhD thesis, Munich University of Technology, Institute of Biomathematics and Biometry, National Research Center for Environment and Health, Munich, Germany, 2005.

[18] H. Führ, L. Demaret, and F. Friedrich. Beyond wavelets: New image representation paradigms. In M. Barni and F. Bartolini, editors, *Document and image coding*. 2006.

[19] D. Geman, S. Geman, and C. Graffigne. Locating texture and object boundaries. In P. Devijer and J. Kittler, editors, *Proceedings of the NATO Advanced Study Institute on Pattern Recognition Theory and Applications*, NASA ASI Series, Berlin, Heidelberg, New York, 1987. Springer Verlag.

[20] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. PAMI*, 6:721–741, 1984.

[21] L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Series in Statistics. Springer-Verlag, New York Berlin Heidelberg, 2002.

[22] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel. *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. Wiley & Sons, New York, 1986.

[23] A. Kempe. *Statistical analysis of discontinuous phenomena with Potts functionals*. PhD thesis, Institute of Biomathematics and Biometry, National Research Center for Environment and Health, Munich, Germany, 2004.

[24] A. Kempe, V. Liebscher, and S. Wichert. A new test for white noise, 2005. in preparation.

[25] H. Künsch. Robust priors for smoothing and image restoration. *Ann. Inst. Statist. Math.*, 46(1):1–19, 1994.

[26] V. Liebscher. TSSEGMENTATION: An R package for segmentation of time series by minimization of penalised least squares or penalised M-estimates, 2005. URL `www.math-inf.uni-greifswald.de/biomathematik/liebscher/misc/`.

[27] V. Liebscher and G. Winkler. A Potts model for segmentation and jump-detection. In V. Benes, J. Janacek, and I. Saxl, editors, *Proceedings S4G International Conference on Stereology, Spatial Statistics and Stochastic Geometry, Prague June 21 to 24, 1999*, pages 185–190, Prague, 1999. Union of Czech Mathematicians and Physicists.

[28] V. Liebscher, O. Wittich, A. Kempe, and G. Winkler. Segmentation of time series: A case study. Preprint, 45 pages, June 2004.

[29] C. R. Loader. Change point estimation using nonparametric regression. *Ann.Stat.*, 24 (4):1667–1678, 1996.

[30] H.-G. Müller. Change-points in nonparametric regression analysis. *Ann.Stat.*, 20(2), 1992.

[31] R. Potts. Some generalized order-disorder transitions. *Proc. Camb. Phil. Soc.*, 48: 106–109, 1952.

[32] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods. A Mathematical Introduction*, volume 27 of *Applications of Mathematics*. Springer Verlag, Berlin, Heidelberg, New York, second edition, 2003. Completely rewritten and revised, Year of publication: 2002.

[33] G. Winkler, V. Aurich, K. Hahn, A. Martin, and K. Rodenacker. Noise reduction in images: Some recent edge-preserving methods. *Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications*, 9(4):749–766, 1999.

[34] G. Winkler and V. Liebscher. Smoothers for discontinuous signals. *J. Nonpar. Statist.*, 14(1-2):203–222, 2002.

[35] G. Winkler, O. Wittich, V. Liebscher, and A. Kempe. Don't shed tears over breaks. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 107(2):57–87, 2005.

[36] O. Wittich, A. Kempe, G. Winkler, and V. Liebscher. Complexity penalized sums of squares for time series': Rigorous analytical results. Schriftenreihe des IBB, 05-1, January 2005. 18 pages.

[37] Y.-C. Yao. Estimating the number of change-points via Schwarz' criterion. *Stat. Probab. Lett.*, 6:181–189, 1988.

[38] Y.-C. Yao and S. Au. Least-squares estimation of a step function. *Sankhya, Ser. A*, 51(3):370–381, 1989.