Information and Inference: A Journal of the IMA (2017) **6**, 225–245
doi:10.1093/imaiai/iaw022
Advance Access Publication on 13 January 2017

Jump-penalized least absolute values estimation of scalar or circle-valued signals

Martin Storath[†]
*Image Analysis and Learning Group, Universität Heidelberg, Im Neuenheimer Feld 205,
69120 Heidelberg, Germany*
[†]Corresponding author: martin.storath@iwr.uni-heidelberg.de

Andreas Weinmann
*Department of Mathematics and Natural Sciences, Darmstadt University of Applied Sciences,
Schöfferstr. 3, 64295 Darmstadt and Institute of Computational Biology, Helmholtz Center Munich,
Ingolstädter Landstraße 1, 85764 Neuherberg, Germany*

and

Michael Unser
*Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne,
Station 17, 1015 Lausanne, Switzerland*

[Received on 23 June 2015; revised on 26 August 2016; accepted on 22 September 2016]

We study jump-penalized estimators based on least absolute deviations which are often referred to as Potts estimators. They are estimators for a parsimonious piecewise constant representation of noisy data having a noise distribution which has heavier tails or which leads to many severe outliers. We consider real-valued data as well as circle-valued data which appear, for instance, as time series of angles or phase signals. We propose efficient algorithms that compute Potts estimators for real-valued scalar as well as for circle-valued data. The real-valued version improves upon the state-of-the-art solver w.r.t. to computational time. In particular for quantized data, the worst case complexity is improved. The circle-valued version is the first efficient algorithm of this kind. As an illustration, we apply our method to estimate the steps in the rotation of the bacterial flagella motor based on real biological data, and to the estimation of wind directions.

Keywords: circle-valued data; jump penalization; least absolute deviation; piecewise constant signal; Potts estimator; step detection.

1. Introduction

Signals or time-series with underlying piecewise constant groundtruth appear in various biological and medical applications; for example, the cross-hybridization of DNA (Snijders *et al.*, 2001; Drobyshev *et al.*, 2003; Hupé *et al.*, 2004), the reconstruction of brain stimuli (Winkler *et al.*, 2005), single-molecule fluorescence resonance energy transfer (Joo *et al.*, 2008), cellular ion channel functionalities (Hotz *et al.*, 2013) and photo-emission spectroscopy (Frick *et al.*, 2014a). A classical engineering example based on piecewise constant functions is crack detection in fracture mechanics. For further applications, we refer to Little & Jones (2011a,b) and Frick *et al.* (2014a), and the references therein. Circle-valued time-series with underlying piecewise constant model appear for example in the rotations of the bacterial flagellar motor; see Sowa *et al.* (2005), Sowa & Berry (2008), Mora *et al.* (2009).

For such data, jump-penalized estimators are powerful tools. A classical instance is the least squares Potts estimator. It measures the complexity of the signal in terms of the number of jumps and penalizes the sums of the squared deviations from data. While the least squares approach is optimally matched to the Gaussian noise model, estimators based on least absolute deviations are more robust to noise with heavier tailed distributions such as Laplacian noise, and they perform better in the presence of outliers. The least absolute values Potts estimator is given by

$$\hat{x} = \arg\min_{x \in X^N} \gamma \, \|\nabla x\|_0 + \sum\nolimits_{n=1}^{N} w_n \, \mathrm{d}(x_n, y_n). \tag{$\mathcal{P}_\gamma$}$$

The first term $\|\nabla x\|_0 = |\{n : x_{n+1} \neq x_n\}|$ measures the complexity of x in terms of the number of jumps, while the second term measures fidelity to data $y \in X^N$, where $\mathrm{d}(x, y)$ is a metric on the space X and w is a vector of positive weights. Here, we consider the spaces $X = \mathbb{R}$ with the standard distance, and $X = \mathbb{S}^1 = \mathbb{T}$ with the arc length distance. The regularization parameter $\gamma > 0$ controls the balance between data deviation and jump penalty.

For scalar data ($X = \mathbb{R}$ with the standard distance), the least absolute values Potts estimator ($\mathcal{P}_\gamma$) takes the familiar form

$$\hat{x} = \arg\min_{x \in \mathbb{R}^N} \gamma \, \|\nabla x\|_0 + \|x - y\|_{1,w}, \tag{1}$$

which we refer to as L^1-Potts estimator. Here the L^1 norm is formed with respect to the weight vector w. Due to their appearance in context of any cyclic data such as phase data or periodic data, we also consider circle-valued data, i.e., the data space $X = \mathbb{S}^1 = \mathbb{T}$ with d being the arc length distance. This results in

$$\hat{x} = \arg\min_{x \in \mathbb{T}^N} \gamma \, \|\nabla x\|_0 + \sum\nolimits_{n=1}^{N} w_n \, \mathrm{arclength}(x_n, y_n). \tag{2}$$

A Potts estimate is a candidate signal that minimizes the corresponding Potts functional; thus the estimates enjoy an optimal tradeoff—optimal with respect to the functional—between data fidelity and complexity measured in terms of the number of jumps. From a Bayesian perspective, Potts functionals are negative posterior loglikelihood functions with an improper prior. Minimal points correspond to the respective maximum a posteriori estimates. These connections are rather of formal than of analytic nature when considering non-finite data spaces; we refer to Winkler *et al.* (2005) for a detailed discussion.

1.1 *Prior and related work*

Potts estimators are named after Renfrey B. Potts who used the prior, respectively, the regularizing term of the functional, in his work on statistical mechanics; see Potts (1952). Related investigations can even be traced back to Ising (1925). Geman & Geman (1984) were one of the first to utilize the Potts model with least squares data fidelity (L^2-Potts estimator) in the context of statistical image processing. In the work of Blake (1983) and Blake & Zisserman (1987), the L^2-Potts model appears as limit case of the nowadays called Blake–Zisserman model. In image processing, the Potts model is also known as piecewise constant Mumford–Shah model after the seminal work of Mumford & Shah (1989). Donoho (1999) considers related bivariate estimators which are called wedgelets. For a thorough account on the history, we refer to Winkler *et al.* (2005) and the references therein.

More recently, Boysen *et al.* (2007, 2009) have given convergence rates and have shown consistency for the L^2-Potts estimator. Wittich *et al.* (2008) have shown uniqueness of the L^2-Potts estimate for almost all input data and idempotence of the corresponding solution operator. In the context of inverse problems, convergence rates have been investigated by Frick *et al.* (2014b), and iterative algorithms by

Fornasier & Ward (2010) and Weinmann & Storath (2015). It is well known that an L^2-Potts estimate can be computed exactly using dynamic programming; see Yao (1984), Mumford & Shah (1985, 1989), Auger & Lawrence (1989), Chambolle (1995), Winkler & Liebscher (2002), Jackson *et al.* (2005). The current state-of-the-art solver is the one of Friedrich *et al.* (2008) which has quadratic time and linear memory complexity. Under the additional assumption that the number of jumps of the underlying signals grows linearly with the signal length, Killick *et al.* (2012) could improve this to a linear time algorithm.

In contrast to the L^2 scenario, significantly less is known on Potts estimators with L^1 data terms. One of the reasons is that the L^1 norm is typically more intricate due to the non-differentiability and the missing relation to the inner product. This stands in contrast to the fact that the L^1 Potts estimator has several advantages over the least squares variant: as already mentioned, L^1 data terms are more robust to noise with heavy tailed distributions and in the presence of outliers. Friedrich *et al.* (2008) observed that they give in practice more robust estimates on the change points (i.e., the jumps) for the considered gene expression data. Remarkably, the L^1-Potts estimator has a certain blind deconvolution property: it reconstructs piecewise constant signals x from given blurred data $y = \kappa * x$, whenever the unknown positive symmetric kernel κ is sufficiently narrow supported; for details see Weinmann *et al.* (2015). Such properties are not shared by the L^2-Potts and L^1-TV estimators. Concerning the computation of minimizers, Friedrich *et al.* (2008) were the first to give an efficient algorithm for computing an estimate of $\mathcal{O}(N^2 \log N)$ time and $\mathcal{O}(N^2)$ memory complexity. The authors of the present article improved this to $\mathcal{O}(N^2)$ time and $\mathcal{O}(N)$ space complexity; see Weinmann *et al.* (2015). Friedrich *et al.* (2008) also consider computing the Potts estimator for all parameters γ simultaneously; they propose an $\mathcal{O}(N^3)$ time and $\mathcal{O}(N^2)$ space algorithm for real-valued data. (We note that the present work deals with quantized and non-quantized data, with values on the real line or on the circle.)

Closely related estimators are the TV estimators which penalize the total variation $\|\nabla x\|_1 = \sum_n |x_{n+1} - x_n|$ instead of the number of jumps. In particular, L^1-TV estimators have been considered in, e.g., Fu *et al.* (2006), Clason *et al.* (2009), Dong *et al.* (2009), Dümbgen & Kovac (2009), Chambolle & Pock (2011), Kolmogorov *et al.* (2016). The Potts prior is a stronger prior in the sense that the underlying signal class of piecewise constant signals is smaller. However, whenever the signal under consideration belongs to this class, Potts estimators are more robust to higher noise levels and enjoy better reconstruction properties. On the flipside, for real-valued data, the computation of a Potts estimate is more challenging because the Potts functional is, in contrast to the L^1-TV functional, not convex.

The Potts estimators discussed above are related to the simultaneous change point estimator (SMUCE) recently proposed by Frick *et al.* (2014a). In contrast to the discussed Potts estimators, which measure the data fidelity in terms of the L^1 or the L^2 norm, SMUCE uses a certain multiscale test statistics. The computational procedure is based on the aforementioned dynamic programming scheme proposed by Friedrich *et al.* (2008) for Potts estimators. A variant that controls the false discovery rate has been proposed by Li *et al.* (2016). Frick *et al.* (2012) consider statistical multiresolution estimators with ℓ^∞ constraints. Xia & Qiu (2015) study an information criterion for jump-penalized estimation. We refer to Little & Jones (2011a,b) and the references therein for further related jump-penalized estimators and algorithms for the reconstruction of piecewise constant real-valued signals.

Besides real-valued data, there is an emerging interest in estimation of circle-valued data. Regression of circular data has been considered by Fisher & Lewis (1983) and by Downs & Mardia (2002), and LASSO/TV type problems by Giaquinta *et al.* (1993), Cremers & Strekalovskiy (2013), Lellmann *et al.* (2013), Weinmann *et al.* (2014), Bergmann *et al.* (2014), Storath *et al.* (2016). This is motivated by their appearance as data spaces in various contexts including phase data, orientation data, as well as nonlinear color spaces. We refer to Fisher (1995) and to Jammalamadaka & SenGupta (2001) for an overview on statistics of circle-valued data. Statistical issues on Riemannian manifolds in general are the topic of Oller & Corcuera (1995), Bhattacharya & Patrangenaru (2003), Bhattacharya & Patrangenaru (2005), Pennec (2006), Fletcher (2013). For manifold-valued data, and thus for circle-valued data in particular, an algorithm for the Potts problem has been proposed by Weinmann *et al.* (2016). However, for circle-valued data, it is not guaranteed that the algorithm converges to a global minimizer. Moreover, the resulting algorithm is of iterative nature and a faster algorithm would be desirable.

1.2 *Contributions*

In this article, we study least absolute values Potts estimators for real-valued as well as for circle-valued data. To our knowledge, Potts estimators for circle-valued data have not been studied yet. Our contributions are as follows: (i) We provide new algorithms which yield global minimizers of the Potts problem for real-valued data (cf. Equation (1)) as well as for circle-valued data (cf. Equation (2)), where the data can be quantized or non-quantized; (ii) we propose a strategy to compute Potts estimates for all regularization parameters $\gamma > 0$ simultaneously with moderate extra effort; and (iii) we provide numerical experiments on synthetic and real-life data.

Concerning (i), the key ingredient is the reduction of the search space to a not *a priori* fixed data dependent set V^N. More precisely, for the case of circle-valued data, we show that we can reduce the search space to V^N, where V contains the unique values of the data y and its antipodal points. The reduction of the problem size allows us to employ the Viterbi-algorithm; see Viterbi (1967), Bellman & Roth (1969), Blake & Zisserman (1987), Felzenszwalb & Zabih (2011). By exploiting the special structure of the Potts penalty as described in Felzenszwalb & Huttenlocher (2006), this leads to an algorithm of complexity $\mathcal{O}(KN)$, where $K \leq 2N$ is the number of elements in the discrete set V. To our knowledge, this is the first exact algorithm for the Potts problem with circle-valued data. For scalar real-valued data, we proceed analogously to obtain a solver of complexity $\mathcal{O}(KN)$. We show that the algorithm has complexity of linearithmic order in probability when the components of the data vector y are independent, discrete random variables which are not too heavy tailed. We stress that the proposed approach is fundamentally different from those of Friedrich *et al.* (2008) and Weinmann *et al.* (2015). The latter ones employ a different dynamic programming scheme, and they do not use a strategy to reduce the search space.

Concerning (ii), we consider the closely related problem of jump-constrained least absolute values approximation, which we call J-jump sparsity problem. We propose an algorithm that solves the J-jump sparsity problem for all possible number of jumps in $\mathcal{O}(KNL)$, where $L \leq N-1$ denotes the number of jumps of the data. Based on this complete scanning, we are able to determine all intervals for γ, where the solution does not change, and all corresponding estimates. Again, our approach relies on a different discrete search space than the $\mathcal{O}(KN^2)$ algorithm of Friedrich *et al.* (2008) for solving for all model parameters γ simultaneously. In particular, our method improves upon the computational complexity of Friedrich *et al.* (2008) for quantized data.

Concerning (iii), we provide a series of numerical experiments on the proposed algorithm. For real-valued (non-quantized) data, we observe that our approach is significantly faster than the previously proposed solver of Weinmann *et al.* (2015). (We recall that the algorithm of Weinmann *et al.*, 2015 is an $O(N^2)$ time and $O(N)$ space algorithm improving the $O(N^2 \log N)$ time and $O(N^2)$ space algorithm of Friedrich *et al.*, 2008.) As a tradeoff, the memory consumption is $\mathcal{O}(KN)$ as opposed to the linear memory complexity of Weinmann *et al.* (2015). Concerning circle-valued data we observe that the proposed algorithm, which is the first one proposed for this problem, yields reasonable runtimes. Further, we apply our method to real data: we estimate the steps in the rotation of the bacterial flagella motor (Sowa *et al.*, 2005) and time series of wind directions.

1.3 *Outline of the article*

In Section 2, we propose efficient algorithms for computing Potts estimates with scalar or circle-valued data. Section 3 deals with the closely related J-jump sparsity problem. In Section 4, we compute solutions of the Potts problem for all parameter simultaneously. Numerical experiments on synthetic data and on real data are conducted in Section 5.

2. Efficient computation of the Potts estimator

We develop efficient solvers for the Potts problem ($\mathcal{P}_\gamma$) for circle-valued and for real-valued data. We first reduce the infinite search space to a finite set, and then utilize the Viterbi algorithm with an acceleration proposed by Felzenszwalb & Huttenlocher (2006).

2.1 *Dimensionality reduction for circle-valued and scalar data spaces*

Our first step towards computing a global minimizer of the Potts functional ($\mathcal{P}_\gamma$) with circle-valued and real-valued data is the reduction of the search space to a finite set. In the following, we use the notation $\mathrm{Val}(y)$ to denote the set of values of a tuple $y \in X^N$, i.e.,

$$\mathrm{Val}(y) = \{v \in X : \text{ there is } 1 \leq n \leq N \text{ s.t. } y_n = v\}.$$

We recall that a (weighted) median μ of $y \in X^N$ is defined as minimizer of the functional $z \mapsto \sum_{n=1}^N w_n\, d(z, y_n)$; see e.g., Fletcher *et al.* (2009). For scalar data, it is well known that the set of values of y contains at least one median. More precisely, there is $\mu \in \arg\min_{z\in\mathbb{R}} \sum_{n=1}^N w_n\, |z - y_n|$ such that $\mu \in \mathrm{Val}(y)$. In the case of circle-valued data, a median is contained in the set of values $\mathrm{Val}(y)$ or the set of antipodal points $\mathrm{Val}(-y)$:

LEMMA 1 Let $y \in \mathbb{T}^N$. The set $V = \mathrm{Val}(y) \cup \mathrm{Val}(-y)$ contains a (weighted) median of y.

Lemma 1 is most probably known. Since we did not find it in the literature, we provide a short proof in the Appendix.

The key to a reduction of the search space is the following statement on solutions of the Potts problem.

THEOREM 2 Let $\gamma > 0$. For real-valued data $y \in \mathbb{R}^N$, we let $V = \mathrm{Val}(y)$; for circle-valued data $y \in \mathbb{T}^N$, we let $V = \mathrm{Val}(y) \cup \mathrm{Val}(-y)$. Then the Potts problem ($\mathcal{P}_\gamma$) has a solution $\hat{x}$ fulfilling $\hat{x} \in V^N$.

The proof is given in the Appendix.

We next discuss how to determine the reduced search space V when data y is given. In many setups, we can *a priori* declare a set U that is only slightly larger than V or, in the best case, equal to V. Then it is convenient to use this set U. For example, we can use $U = \{0, \ldots, 255\}$ for an 8-bit signal. If there is no such set we have to determine V algorithmically. A simple method to do so is based on sorting of y. Then, duplicate elements are adjacent and can be easily removed. However, this procedure may exceed the complexity of the subsequent step (see Section 2.2), because general purpose sorting algorithms are $\mathcal{O}(N \log N)$. To avoid sorting, we utilize the following simple procedure: We create an empty list V. Then we iterate over the components of y, that is, for $n = 1, \ldots, N$. If y_n is not in the list, we append it to V. In case of circle-valued data, we additionally append the antipodal points. The procedure needs at most $K = |V|$ comparisons at each step, thus the procedure is $\mathcal{O}(KN)$.

2.2 *A Viterbi-type algorithm for the reduced problem*

Theorem 2 allows us to reduce the infinite search spaces $\mathbb{R}^N$ and $\mathbb{T}^N$ in $(\mathcal{P}_\gamma)$ to the finite set V^N, which is specified in that theorem. Thus, it remains to solve

$$\arg\min_{x \in V^N} \gamma \, \|\nabla x\|_0 + \sum\nolimits_{n=1}^{N} w_n \, d(x_n, y_n). \tag{3}$$

To this end, we use dynamic programming. Recall that the basic idea of dynamic programming is to solve a complex problem by reducing it to smaller subproblems; see Bellman (1957). In the following, we denote the cardinality of V by K. It is useful to equip the set V with indices, i.e., $V = \{v_1, \ldots, v_K\}$. (Note that any ordering is valid.)

We utilize a dynamic programming scheme developed by Viterbi (1967); see also Forney (1973). Related algorithms have been proposed by Bellman & Roth (1969) and Blake & Zisserman (1987). Here, we review a special instance of the Viterbi algorithm proposed by Felzenszwalb & Huttenlocher (2006); see also Felzenszwalb & Zabih (2011). Assume that we aim at minimizing an energy functional of the form

$$E(x_1, \ldots, x_N) = \gamma \sum\nolimits_{n=1}^{N-1} h(x_n, x_{n+1}) + \sum\nolimits_{n=1}^{N} w_n \, d(x_n, y_n), \tag{4}$$

where $x_1, \ldots, x_N$ can take values in a finite set V and where h is an energy functional on V^2. The Viterbi algorithm solves this problem in two steps: tabulation of energies and reconstruction by backtracking. For the tabulation, the starting point is the table $B^1 \in \mathbb{R}^K$ given by

$$B^1_k = w_1 d(v_k, y_1) \quad \text{for } k = 1, \ldots, K.$$

We successively compute for $n = 2, \ldots, N$ the tables $B^n \in \mathbb{R}^K$ which are given by

$$B^n_k = w_n d(v_k, y_n) + \min_l \{B^{n-1}_l + \gamma \, h(v_k, v_l)\}, \quad \text{for } k = 1, \ldots, K. \tag{5}$$

The entry B^n_k represents the energy of a minimizer on data $(y_1, \ldots, y_n)$ whose endpoint is equal to v_k. For the backtracking step, it is convenient to introduce an auxiliary tuple $l \in \mathbb{N}^N$ which stores minimizing indices. We initialize the last entry of l by $l_N = \arg\min_k B^N_k$. Then we successively compute the entries of l for $n = N-1, N-2, \ldots, 1$ by

$$l_n = \arg\min_k B^n_k + \gamma \, h(v_k, v_{l_{n+1}}). \tag{6}$$

Eventually, we reconstruct a minimizer $\hat{x}$ from the indices in l by $\hat{x}_n = v_{l_n}$, for $n = 1, \ldots, N$. The result $\hat{x}$ is a global minimizer of the energy (4); see Felzenszwalb & Zabih (2011).

As proposed by Felzenszwalb & Huttenlocher (2006), the special structure of the Potts prior can be exploited to streamline the computation. First note that the reduced Potts problem (3) corresponds to the functional defined by $h(v,w) = 1$ for $v \neq w$ and $h(v,v) = 0$. This allows to simplify the second term in (5) to

$$\min_l \{B_l^{n-1} + \gamma\, h(v_k, v_l)\} = \min\{\gamma + \min_l B_l^{n-1}; B_k^{n-1}\}.$$

It follows that the table B^n can be filled by the two-step procedure

$$\begin{cases} z = \min_l B_l^{n-1}, \\ B_k^n = w_n d(y_n, v_k) + \min\{\gamma + z; B_k^{n-1}\}, \quad \text{for } k = 1, \ldots, K. \end{cases} \tag{7}$$

Now z can be precomputed and reused for all table entries B_k^n for $k = 1, \ldots, K$. Since both lines of (7) are $\mathcal{O}(K)$, the computation of the table B^n is $\mathcal{O}(K)$. Hence, this algorithm is $\mathcal{O}(KN)$.

2.3 *Complete solver for the Potts problem*

The computation of the global minimizer of the Potts problem $(\mathcal{P}_\gamma)$ is performed in two steps: computation of the reduced search space V followed by the above variant of the Viterbi algorithm. A pseudocode of the complete algorithm is given in Algorithm 1. We record the following:

Theorem 3 Algorithm 1 computes a global minimizer of the Potts problem $(\mathcal{P}_\gamma)$ with scalar or circle-valued data in $\mathcal{O}(KN)$.

The proof is given in the Appendix.

For the Potts problem with scalar data, a solver of complexity $\mathcal{O}(N^2 \log N)$ has been proposed by Friedrich *et al.* (2008) and improved to $\mathcal{O}(N^2)$ by Weinmann *et al.* (2015). These methods use also the dynamic programming principle, but with a fundamentally different tabulation scheme. It is based on computing the median deviations on all $\mathcal{O}(N^2)$ intervals. In contrast, the tabulation in Algorithm 1 is based on the distances of single data points y_n to all the K median candidates. For the Potts problem with circle-valued data, no efficient solver has been proposed yet to the best of our knowledge.

2.4 *Algorithmic complexity for quantized data*

In general, the number of elements of V, denoted by K, is less than or equal to the number of data points N for scalar data and less than or equal to $2N$ for circle-valued data. Thus the worst case complexity of Algorithm 1 is $\mathcal{O}(N^2)$. In many situations, K is significantly smaller than N. For quantized data on the unit circle, the number of discrete values K is bounded. Thus we obtain linear complexity in this case.

In the case of quantized data on the real line, the number of values might not always be bounded. However, we can give a bound for the following typical situation. Let f be a time-continuous piecewise constant function on the interval $[0,1]$ with finitely many jumps, denoted by $\|\nabla f\|_0$. We further assume that f is a càdlàg function, which means that the limits $f(t+)$ and $f(t-)$ exist and $f(t+) = f(t)$ for all $t \in [0,1]$. Let $\bar{y}^N$ be the (point or integral) sampling of f at density $1/N$; that is, $\bar{y}_n^N = f\left(\frac{n}{N+1}\right)$ or $\bar{y}_n^N = \frac{1}{N}\int_{\frac{n-1}{N}}^{\frac{n}{N}} f(t)\,\mathrm{d}t$. Data at sampling density $1/N$ is given by

$$y_n^N = \bar{y}_n^N + \eta_n^N, \tag{8}$$

Algorithm 1: Solver for the Potts problem ($\mathcal{P}_\gamma$) with real-valued or circle-valued data

```
Input: $y \in \mathbb{R}^N$ or $y \in \mathbb{T}^N$: data; $\gamma > 0$: model parameter; $w \in (\mathbb{R}_0^+)^N$: weights;
Output: Global minimizer $\hat{x}$ of ($\mathcal{P}_\gamma$);
begin
    /* Search space reduction                                                    */
    Determine reduced search space $V$ (see Section 2.1);
    /* Tabulation                                                                */
    for $k \leftarrow 1$ to $K$ do
        $B_k^1 \leftarrow w_1 d(v_k, y_1)$;
    end
    for $n \leftarrow 2$ to $N$ do
        $z \leftarrow \min_{k=1,\dots,K} B_k^{n-1}$;
        for $k \leftarrow 1$ to $K$ do
            $B_k^n \leftarrow w_n d(v_k, y_n) + \min(z + \gamma, B_k^{n-1})$;
        end
    end
    /* Backtracking                                                              */
    $l \leftarrow \arg\min_{k=1,\dots,K} B_k^N$;
    $\hat{x}_n \leftarrow v_l$;
    for $n \leftarrow N-1, N-2, \dots, 1$ do
        $B_l^n \leftarrow B_l^n - \gamma$;
        $l \leftarrow \arg\min_{k=1,\dots,K} B_k^n$;
        $\hat{x}_n \leftarrow v_l$;
    end
    return $\hat{x}$;
end
```

where η^N is an N-dimensional vector of independent random variables. Since

$$K = |\mathrm{Val}(y_n^N)| \le |\mathrm{Val}(\bar{y}_n^N)|\,|\mathrm{Val}(\eta_n^N)| \le C_f |\mathrm{Val}(\eta_n^N)|$$

with the constant $C_f = 2(\|\nabla f\|_0 + 1)$, the number of expected values is bounded by the number of expected values of the noise η. If the supports of the distribution of the components of η are uniformly bounded with respect to n, N, then K is bounded and we obtain linear complexity of Algorithm 1. For unbounded supports, we get a slightly higher algorithmic complexity.

PROPOSITION 4 Let $C > 0$ and $0 < p < 1$. If η^N is an N-dimensional vector of independent random variables, where each component follows an integer-valued probability distribution satisfying the exponential decay condition $P(\eta_n^N = k) \le Cp^{|k|}$, then $|\mathrm{Val}(\eta^N)| \in \mathcal{O}_p(\log N)$.

The proof is given in the Appendix. Examples of distributions that satisfy the hypothesis of Proposition 4 are the discrete Laplace distribution, the discrete Gaussian distribution and the geometric distribution. We summarize the complexity of Algorithm 1 for the discussed data in Table 1.

REMARK 1 The statement of Proposition 4 remains true when each of the probability distributions of the independent random variables η_n^N are discrete with a global minimum separation distance, i.e., there

TABLE 1 *Complexity of the proposed solver for the least absolute values Potts problem (Algorithm 1) for quantized and non-quantized, real-valued and circle-valued data. The last two rows for quantized real-valued data refer to data of the form (8).*

	Circle-valued data	Real-valued data
Non-quantized data	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
Quantized data	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$ general case $\mathcal{O}_p(N \log N)$ for noise with exp. decay (Proposition 4) $\mathcal{O}(N)$ for bounded noise

is $\varepsilon > 0$ such that any pair of neighboring values a, b in the range of any random variable η_n^N fulfill $|a - b| > \varepsilon$. In this case, the decay condition of Proposition 4 reads $P(\eta_n^N = a) \leq Cp^{|a|}$, where a is in the range of the random variable η_n^N.

3. The *J*-jump sparsity problem

Next we consider the *J-jump sparsity problem* given by

$$\underset{\|\nabla x\|_0 \leq J}{\operatorname{argmin}} \sum_{n=1}^{N} w_n \, \mathrm{d}(x_n, y_n), \qquad (\mathcal{P}_J)$$

where $J \in \mathbb{N}_0$. Its solution will be useful for solving the Potts problem $(\mathcal{P}_\gamma)$ for all parameters simultaneously (see next section). Apart from this, $(\mathcal{P}_J)$ is an interesting problem in its own right which has applications, for example, in minimum error quantization; see Bruce (1965).

3.1 *Basic properties of the J-jump sparsity problem*

In the following, we denote by L the number of jumps of the data y; that is, $L = \|\nabla y\|_0$. For $J \geq L$, the J-jump sparsity problem $(\mathcal{P}_J)$ has the trivial solution y. Thus, we can restrict ourselves to the case of J not being larger than L.

Although the J-jump sparsity problem $(\mathcal{P}_J)$ and the Potts problem $(\mathcal{P}_\gamma)$ are closely related, they are not fully equivalent in the sense that a minimizer of $(\mathcal{P}_J)$ may not appear as minimizer of $(\mathcal{P}_\gamma)$, no matter of the value of γ. To see this, consider the scalar data $y = (0, 1, 0)$ and as data term the Euclidean distance with uniform weights. The possible solutions for $J = 1$ are the one-jump solutions of the form $(0, a, a)$, $(a, a, 0)$, $a \in [0, 1]$, and the zero-jump solution $(0, 0, 0)$. However, one-jump solutions cannot be minimizers of the Potts functional: their Potts functional values are equal to $\gamma + 1$, but this is for all $\gamma > 0$ strictly higher than the functional value of the zero jump solution which is equal to 1. An analogous example can be given for circle-valued data.

The above example also shows that the J-jump sparsity problem can have solutions with $(J - 1)$ jumps. Next, we argue that a solution of the J-jump sparsity problem cannot have less than $(J - 1)$ jumps. Assume that a solution x^J has $(J - 2)$ or fewer jumps. Then we find an index n such that $x_n^J \neq y_n$. Setting x_n^J equal to y_n adds at most two jumps. Thus, we have still at most J jumps. But the approximation error of that candidate is lower than that of x^J, which contradicts the fact that x^J minimizes the J-jump sparsity problem.

3.2 *An efficient solver based on dynamic programming*

As with the Potts problem, the key to a fast solver is a reduction of the search space to the finite set V^N:

THEOREM 5 Let $J \in \mathbb{N}_0$ and let V be as in Theorem 2. The J-jump sparsity problem ($\mathcal{P}_J$) has a solution $\hat{x}$ that fulfills $\hat{x} \in V^N$.

The proof is an obvious modification of the proof of Theorem 2.

The computational solution is also similar to the solver of the Potts problem. The main difference is that we now require JN tables of length K instead of N tables of length K. A table entry $B_k^{n,j}$ ($1 \leq n \leq N$, $1 \leq j \leq J$, $1 \leq k \leq K$) stores the approximation energy of the optimal j-jump solution x for data $(y_1, \dots, y_n)$ with $x_n = v_k$. The first J tables $B^{1,j}$ are given by $B_k^{1,j} = w_1 d(v_k, y_1)$, for $j = 0, \dots, J$. Using the dynamic programming principle, we fill the tables $B^{n,j}$ successively for $n = 1, \dots, N$ and for $j = 0, \dots, J$ by

$$B_k^{n,j} = \min \left\{ z; w_n d(v_k, y_n) + B_k^{n-1,j} \right\},$$

where $z = \min_{l=1,\dots,K} B_l^{n-1,j-1}$. As z does not depend on k, it can be computed separately. Filling the table $B^{n,j}$ requires K steps; hence, the total complexity for the tabulation is $\mathcal{O}(NKJ)$. For the backtracking step, we first determine the minimizer l_N of the table $B^{N,J}$, i.e., $l_N = \arg\min_k B_k^{N,J}$. (As for the Potts problem, the tuple $l \in \mathbb{N}^N$ stores minimizing indices.) We further initialize the number of remaining jumps by $i_N = J$. For $n = N-1, \dots, 1$, we decide—based on the tables—whether it is advantageous to take a jump or to continue with the same value via

$$(l_{n-1}, i_{n-1}) = \operatorname*{argmin}_{(k,j)\in\{(1,i_n-1),\dots,(K,i_n-1),(l_n,i_n)\}} B_k^{n,j}.$$

The procedure terminates if $i_n = 0$, and then we set $l_1 = \cdots = l_n$. Having computed the minimizing indices l_n, we obtain a minimizer $\hat{x}$ of the J-jump sparsity problem by $\hat{x}_n = v_{l_n}$, for $n = 1, \dots, N$. The complexity of the backtracking step is $\mathcal{O}(KN)$.

A minor modification of the backtracking step allows us to compute the minimizers for the jump constrained problem for all parameters $J' \leq J$ without having to build up new tables. We just have to initialize the backtracking step with $i_N = J'$ instead of $i_N = J$. In particular, if we want to compute minimizers for all possible number of jumps $J = 0, \dots, L$, we only need compute the tabulation for $J = L$ and perform $L + 1$ backtracking steps with the initializations $i_N = 0, \dots, L$. This amounts to the complexity $\mathcal{O}(KNL)$.

We eventually summarize the results of the above derivation:

THEOREM 6 The above algorithm computes a global minimizer of ($\mathcal{P}_J$) in $\mathcal{O}(KNJ)$. Computing minimizers for all $J = 0, \dots, L$ simultaneously is $\mathcal{O}(KNL)$.

4. Solving the Potts problem for all parameters simultaneously

We can solve the Potts problem ($\mathcal{P}_\gamma$) for all parameters $\gamma > 0$ simultaneously with a moderate effort. To that end, we exploit the following relation between the Potts and the J-jump sparsity problems.

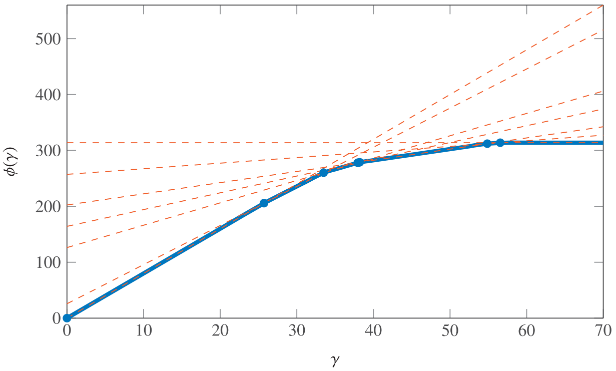


FIG. 1. The function $\phi(\gamma)$ (blue solid line) is the largest minorant of the affine functions $\gamma J + \epsilon_J$ for $J = 0, \ldots, L$ (red dashed lines).

LEMMA 7 Let $\gamma > 0$ and let $\hat{x}$ be a solution of the Potts problem ($\mathcal{P}_\gamma$). Then $\hat{x}$ is also a solution of the jump constrained problem ($\mathcal{P}_J$) with parameter $J = \|\nabla \hat{x}\|_0$.

The proof is given in the Appendix. The example in the first paragraph of Section 3 shows that the opposite direction of Lemma 7 is not true in general. Now assume that we have computed for each $J = 0, \ldots, L$, a minimizer for the J-jump sparsity problem, which we denote by x^J. (Recall that this can be done in $\mathcal{O}(KNL)$ by Theorem 6.) Lemma 7 tells us that the set $\{x^0, \ldots, x^L\}$ already contains a solution of ($\mathcal{P}_\gamma$) for each $\gamma > 0$. This implies that we obtain a solution $\hat{x}$ of ($\mathcal{P}_\gamma$) by scanning through the J-jump solutions; that is,

$$\hat{x} \in \underset{x \in \{x^0, \ldots, x^L\}}{\operatorname{argmin}} \; \gamma \|\nabla x\|_0 + \sum\nolimits_{n=1}^{N} w_n \, d(x_n, y_n).$$

It remains to identify the connection between γ and J. We denote the minimal Potts functional value for some $\gamma > 0$ by $\phi(\gamma)$, and we rewrite it as

$$\phi(\gamma) = \min_x P_\gamma(x) = \min_{x \in \{x^0, \ldots, x^L\}} \gamma \, \|\nabla x\|_0 + \sum\nolimits_{n=1}^{N} w_n \, d(x_n, y_n) = \min_{J=0,\ldots,L} \gamma J + \epsilon_J, \tag{9}$$

where $\epsilon_J = \sum_{n=1}^{N} w_n \, d(x_n^J, y_n)$ is the approximation error of a J-jump solution x^J. Of even greater interest is the function that maps γ to the minimizing argument J:

$$\hat{J}(\gamma) = \underset{J=0,\ldots,L}{\operatorname{argmin}} \, \gamma J + \epsilon_J. \tag{10}$$

With the help of $\hat{J}$, we obtain a minimizer of ($\mathcal{P}_\gamma$) by $\hat{x} = x^{\hat{J}(\gamma)}$.

Let us now explain how to compute the mappings ϕ and $\hat{J}$ given by (9) and (10) efficiently. Being the pointwise infimum of $L + 1$ affine linear functions, ϕ is concave and piecewise linear (with at most $L + 1$ pieces). (It is the largest concave minorant of these affine linear functions.) The graph of ϕ is a polygon which is determined by at most L points; see Fig. 1. Let us denote the affine function $\gamma J + \epsilon_J$ by g^J. We first compute the intersection of g^0 and g^1 and store that point on a stack. We proceed inductively as follows. Assume that we have computed the concave minorant, ϕ^J, of the first J affine functions. Also assume that we have stored the determining nodes on a stack S, where at the top of the stack is the leftmost node. In the step $J+1$, we compute the intersection of the line g^{J+1} with ϕ^J. Since the slope of g^{J+1} is larger than all slopes of ϕ^J there is at most one point of intersection. We first delete the nodes of ϕ_J that lie above the line g^J; that is, we pop the stack until we encounter a point which is below the new line. Then we compute the intersection of the line through the point we have popped last and the point at top of the stack. We push that point to the stack. We proceed until no more lines have to be added. The points in the stack then determine the graph of ϕ. (The slope to ∞ is equal to 0 and the slope to 0 is given by the number of jumps of the data, $\|\nabla y\|_0$.) The mapping $\hat{J}$ is determined by the slopes of ϕ.

We eventually summarize the results of this section:

Theorem 8 There is an $\mathcal{O}(KNL)$ algorithm that computes minimizers of $(\mathcal{P}_\gamma)$ for all parameters $\gamma > 0$ simultaneously.

5. Numerical experiments

We present numerical results on synthetic data with known ground truth, and we show results on real data. The experiments were conducted on a desktop computer (Intel Xeon E5, 3.5 GHz, 32 GB RAM). The proposed algorithms were implemented in Matlab.

5.1 *Synthetic real-valued data*

An extensive numerical study on the denoising performance of the L^1-Potts estimator has been given in earlier works by Friedrich *et al.* (2008) and Weinmann *et al.* (2015). Therefore, we here focus on comparison to the runtimes of the solver by Weinmann *et al.* (2015).[1] (Recall that this solver improved upon the first solver of Friedrich *et al.*, 2008 both in time and memory complexity.) As a basis for our experiment we use the signal previously employed in Fig. 1 of Weinmann *et al.* (2015) as ground truth; it has eight jumps and a dynamic range of [0, 1]. We corrupted it by Laplacian noise of standard deviation $\sigma = 0.1$. As a result, the final signal has $K = N$ different values. In Fig. 2, we show the average runtime with respect to 10 different realizations of the noise. We observe that Algorithm 1 is about twice as fast as the state-of-the-art solver for general non-quantized data. For quantized data, the proposed algorithm becomes significantly faster as the length of the data increases. Thus, it might be reasonable to quantize the data to a desired precision before applying the proposed solver when dealing with very large data. We note that the runtimes of the algorithm of Weinmann *et al.* (2015) and the proposed algorithm for non-quantized data are by construction virtually independent of the number of jumps of the underlying signal.

In Fig. 3, we compare the runtimes for different noise distributions, where data is quantized to three decimals. Uniform noise is bounded, so the runtime grows linearly in N. Laplacian noise leads to linearithmic scaling as it satisfies the hypothesis of Proposition 4. Even for Cauchy noise, the complexity grows significantly slower than quadratic, although it has infinite variance and its probability density has very slow decay. In all cases, the computational time is significantly lower than the one of the benchmark method of Weinmann *et al.* (2015), which exhibits almost no dependence of the runtimes on the noise.

[1] The reference implementation of the solver by Weinmann *et al.* (2015) is available at http://pottslab.de, accessed December 8, 2016.

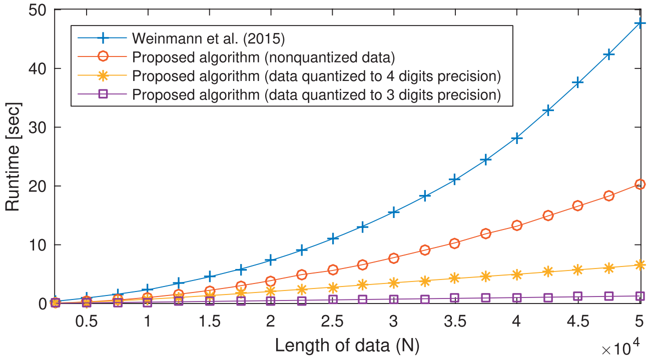


FIG. 2. Runtime of our algorithm for the Potts problem with real-valued data. The proposed method is faster than the state-of-the-art solver of Weinmann *et al.* (2015). With a prior quantization, we are able to process even very large data within a few seconds.

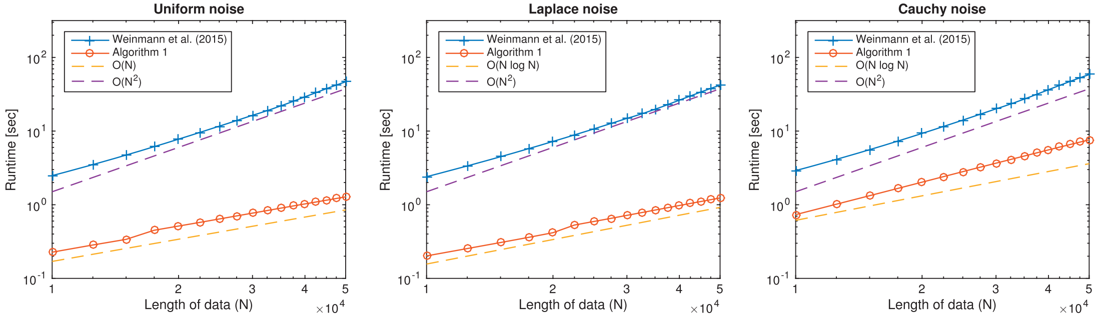
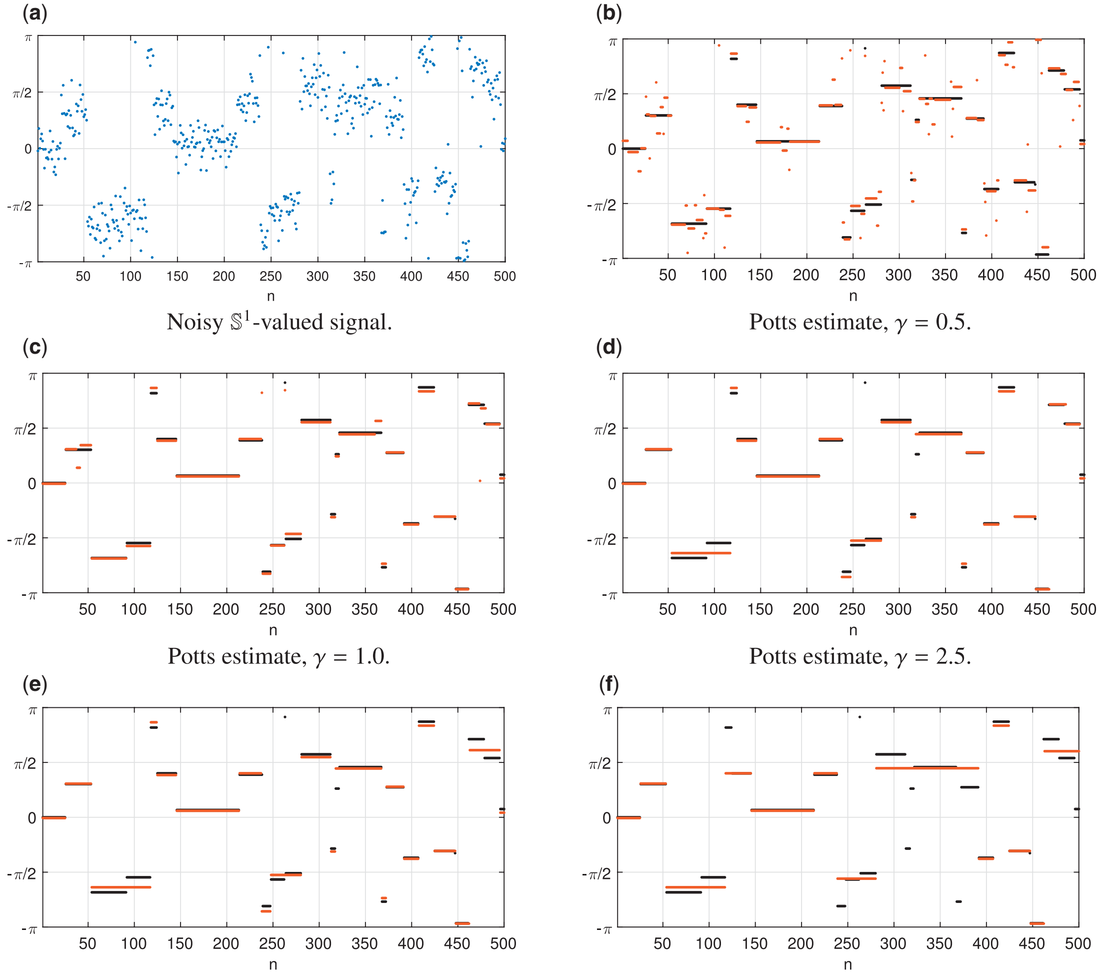


FIG. 3. Runtime of our algorithm for the Potts problem for data with different noise distributions. All signals are quantized to three decimals. The runtimes are averages over 100 realizations. We observe that the runtime grows linearly in N for uniform noise. Laplacian noise leads to linearithmic scaling as suggested by Proposition 4. Even for Cauchy noise, the computational effort grows significantly slower than quadratic.

5.2 *Synthetic circle-valued data*

We illustrate the performance of the Potts estimator for denoising synthetic circle-valued signals (with known ground truth). To this end, we create random jump-sparse signals as follows: We create a compound Poisson distributed random vector $s \in \mathbb{R}^N$; that is, $s_n = 0$ with probability $\mathrm{e}^{-\lambda}$ and s_n is uniformly distributed in $[-a, a]$ with probability $1 - \mathrm{e}^{-\lambda}$; see Unser & Tafti (2014). Here we use $\lambda = 0.05$ and $a = 1/2$. The (true) signal $\bar{y}$ is given as the summation process of the innovation s interpreted as angle; that is, $\bar{y}_n = \exp\left(i \sum_{j=1}^{n} s_j\right)$. We add noise to the phase by $y = \bar{y}\exp(i\eta)$, where η is a Laplacian distributed random vector with standard deviation $\sigma = 0.4$. In Fig. 4, we report a realization of the random signal and its Potts estimate for different model parameters γ. In Fig. 5, we show the deviation of the estimates to the ground truth with respect to the sum over the pointwise arc length distances, and with respect to the number of jumps in dependance of the model parameter γ. (Recall that the Potts estimate for all model parameters γ can be computed with moderate effort by the algorithm described in Section 4.) Figure 6 shows the results for a different noise distribution, namely a wrapped Gaussian distribution with $\sigma = 0.3$.



Noisy $\mathbb{S}^1$-valued signal.

Potts estimate, $\gamma = 0.5$.

Potts estimate, $\gamma = 1.0$.

Potts estimate, $\gamma = 2.5$.

Potts estimate, $\gamma = 5.0$.

Potts estimate, $\gamma = 10.0$.

FIG. 4. (a) A circle-valued signal of length $N = 500$ in angular representation corrupted with wrapped Laplacian noise with $\sigma = 0.4$. (b)–(f) Potts estimate visualized for various values of the model parameter γ, plotted as red line. The ground truth is drawn in black. (See also Fig. 5 for the errors in dependance of the γ parameter.)

Eventually we note that the runtimes for circle-valued data are of the same order of magnitude as those of the real-valued case for boundedly quantized and unquantized data, respectively.

5.3 *Results on real-world data*

Next we apply our method to real-world data. First, we look at the estimation of steps in the rotation of the bacterial flagellar motor. The bacterial flagellar motor is a rotary molecular machine that is embedded in the bacterial cell envelope (Sowa *et al.*, 2005; Sowa & Berry, 2008). It propels many species of swimming bacteria. Sowa *et al.* (2005) observed steps in the rotation of the flagellar motor over time;

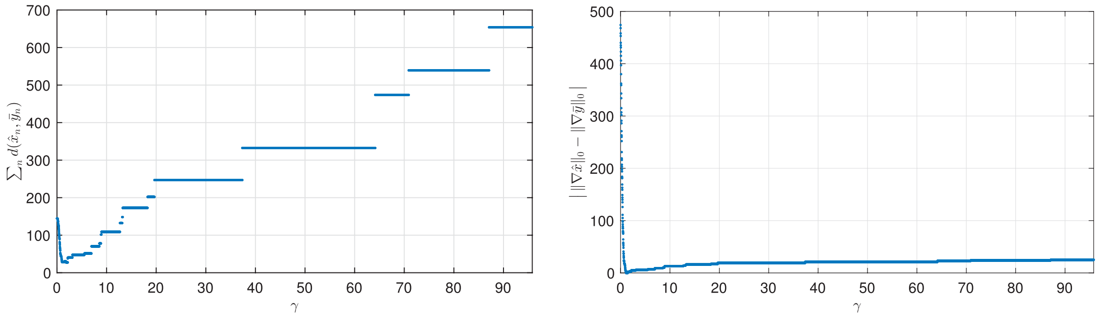
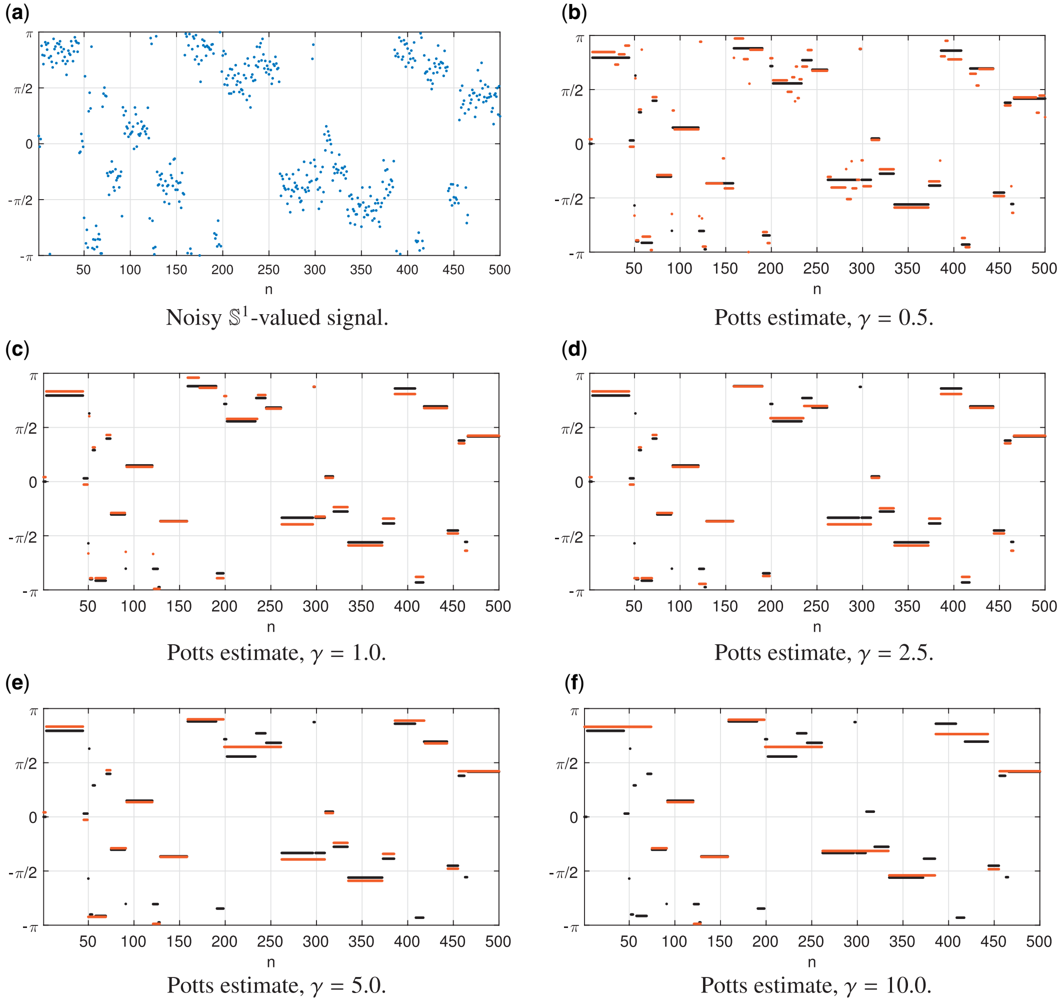


FIG. 5. The plots illustrate the total absolute deviation of the Potts estimate $\hat{x}$ and to ground truth $\bar{y}$ (left) and the difference w.r.t. to the number of jumps in dependance of model parameter γ (right). The graphs correspond to the signal in Fig. 4.



Noisy $\mathbb{S}^1$-valued signal.

Potts estimate, $\gamma = 0.5$.

Potts estimate, $\gamma = 1.0$.

Potts estimate, $\gamma = 2.5$.

Potts estimate, $\gamma = 5.0$.

Potts estimate, $\gamma = 10.0$.

FIG. 6. (a) A circle-valued signal of length $N = 500$ in angular representation corrupted with wrapped Gaussian noise with $\sigma = 0.3$. (b)–(f) Potts estimate visualized for various values of the model parameter γ, plotted as red line. The ground truth is drawn in black.

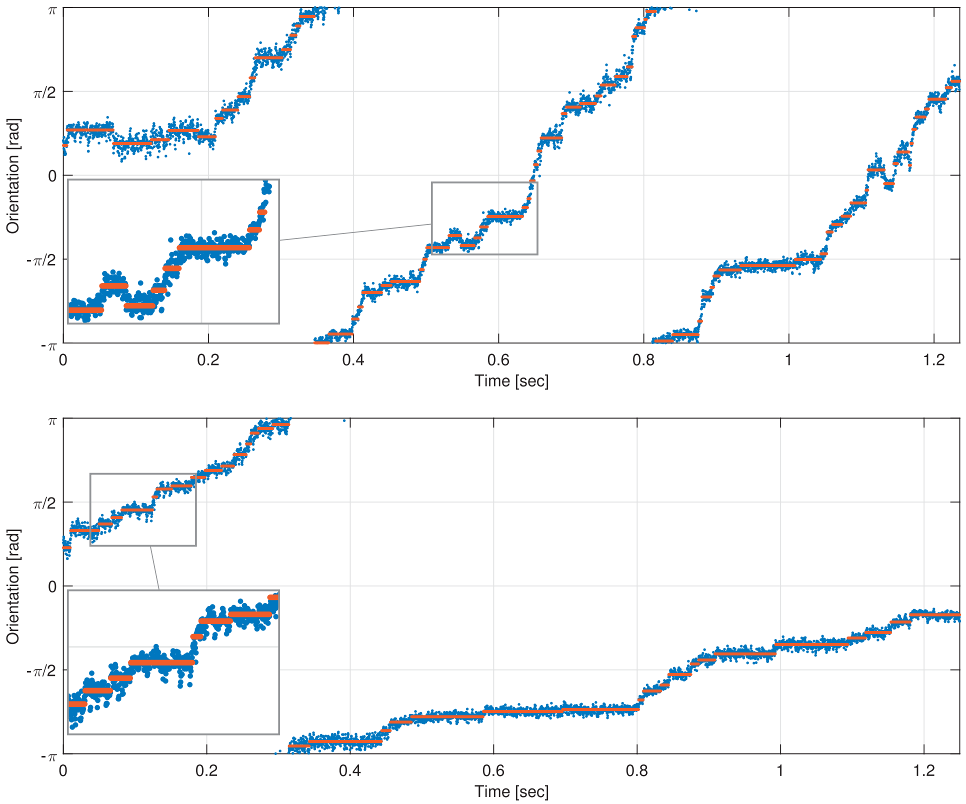


FIG. 7. The blue points represent the angle of the bacterial flagellar motor over time. The red line is the least absolute values Potts estimate. (Original data by courtesy of Sowa *et al.*, 2005.)

in average, they found approximately 26 steps per revolution. The present data set is a time series of the angular position of the flagellar motor. The data was acquired using back-focal plane interferometry. The measurement uncertainty of the data acquisition results in a slightly blurred signal. In view of the built-in deconvolution property of the L^1-Potts functional (cf. Theorem 4.4 in Weinmann *et al.*, 2015), we expect that a slight blur does not negatively affect the performance of the Potts estimator. In Fig. 7, we report the estimate for the model parameter $\gamma = 1$.

Our second real-life data set consists of wind directions at the station WPOW1 (West Point, WA) recorded every hour in the year 2013.[2] The Potts estimate facilitates to identify the time intervals of approximately constant wind direction. The data is given quantized to integer angles in degrees, i.e., $K = 360$. In Fig. 8, we report the result for the model parameter $\gamma = 10$.

6. Conclusion

We have studied the Potts estimator based on least absolute deviations for real-valued data and for circle-valued data. We have proposed exact solvers for the Potts problem with real-valued and circle-valued

[2] Data available at http://www.ndbc.noaa.gov/historical_data.shtml, accessed December 8, 2016.

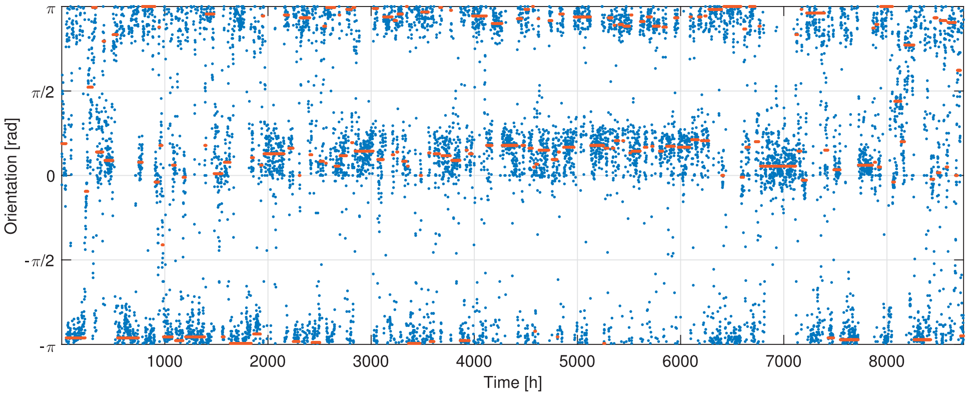


FIG. 8. Wind directions at Station WPOW1 (West Point, WA) recorded every hour in the year 2013. (The data is given quantized to $K = 360$ angles.) The red line is the least absolute values Potts estimate for circle-valued data.

data. Here, the key observation was that solutions take values in a finite set (also for non-quantized data) which allowed us to employ a Viterbi-type dynamic program as proposed in Felzenszwalb & Huttenlocher (2006). Furthermore, we have obtained an algorithm for solving the Potts problem for all parameters γ simultaneously which was based on solving the related J-jump sparsity problem. The proposed algorithm for real-valued data improves upon the state-of-the-art solver with respect to computational time. The circle-valued variant is the first exact solver for the circle-valued Potts problem to our knowledge. We have illustrated the practical utility for the estimation of two real-life data sets: estimation of steps in the rotation of the bacterial flagellar motor and estimation of time series of wind directions. An interesting topic of future research are strategies for choosing the model parameter.

Acknowledgements

We would like to thank Yoshiyuki Sowa, Alexander Rowe, Mark Leake, Toshiharu Yakushi, Michio Homma, Akihiko Ishijima and Richard Berry for providing the data set of the bacterial flagellar motor. We further would like to thank the anonymous reviewers for their valuable comments and suggestions.

Funding

German Research Foundation (STO1126/2-1, WE5886/4-1 to M.S., A.W.). Helmholtz Association within the young investigator group VH-NG-526 to A.W. European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no. 267439 to M.U.

REFERENCES

AUGER, I. & LAWRENCE, C. (1989) Algorithms for the optimal identification of segment neighborhoods. *Bull. Math. Biol.*, **51**, 39–54.

BELLMAN, R. (1957) *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press.

BELLMAN, R. & ROTH, R. (1969) Curve fitting by segmented straight lines. *J. Am. Stat. Assoc.*, **64**, 1079–1084.

Bergmann, R., Laus, F., Steidl, G. & Weinmann, A. (2014). Second order differences of cyclic data and applications in variational denoising. *SIAM J Imaging Sci.*, **7**, 2916–2953.

Bhattacharya, R. & Patrangenaru, V. (2003) Large sample theory of intrinsic and extrinsic sample means on manifolds I. *Ann. Stat.*, **31**, 1–29.

Bhattacharya, R. & Patrangenaru, V. (2005) Large sample theory of intrinsic and extrinsic sample means on manifolds II. *Ann. Stat.*, **33**, 1225–1259.

Blake, A. (1983) The least-disturbance principle and weak constraints. *Pattern Recognit Lett.*, **1**, 393–399.

Blake, A. & Zisserman, A. (1987) *Visual Reconstruction*. Cambridge: MIT Press.

Boysen, L., Kempe, A., Liebscher, V., Munk, A. & Wittich, O. (2009) Consistencies and rates of convergence of jump-penalized least squares estimators. *Ann. Stat.*, **37**, 157–183.

Boysen, L., Liebscher, V., Munk, A. & Wittich, O. (2007) Scale space consistency of piecewise constant least squares estimators: another look at the regressogram. *Lecture Notes-Monograph Series, vol. 55, Asymptotics: Particles, Processes and Inverse Problems: Festschrift for Piet Groeneboom*, Beachwood, Ohio, USA: Institute of Mathematical Statistics, pp. 65–84.

Bruce, J. (1965) Optimum quantization. *Technical Report 429*, Cambridge, Massachusetts: Massachusetts Institute of Technology.

Chambolle, A. (1995) Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations. *SIAM J. Appl. Math.*, **55**, 827–863.

Chambolle, A. & Pock, T. (2011) A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, **40**, 120–145.

Clason, C., Jin, B. & Kunisch, K. (2009) A duality-based splitting method for ℓ^1-TV image restoration with automatic regularization parameter choice. *SIAM J. Sci. Comput.*, **32**, 1484–1505.

Cremers, D. & Strekalovskiy, E. (2013) Total cyclic variation and generalizations. *J. Math. Imaging Vis.*, **47**, 258–277.

Dong, Y., Hintermüller, M. & Neri, M. (2009) An efficient primal-dual method for L^1 TV image restoration. *SIAM J. Imaging Sci.*, **2**, 1168–1189.

Donoho, D. (1999) Wedgelets: nearly minimax estimation of edges. *Ann. Stat.*, **27**, 859–897.

Downs, T. & Mardia, K. (2002) Circular regression. *Biometrika*, **89**, 683–698.

Drobyshev, A., Machka, C., Horsch, M., Seltmann, M., Liebscher, V., de Angelis, M. & Beckers, J. (2003) Specificity assessment from fractionation experiments (safe): a novel method to evaluate microarray probe specificity based on hybridisation stringencies. *Nucleic Acids Res.*, **31**, 1–10.

Dümbgen, L. & Kovac, A. (2009) Extensions of smoothing via taut strings. *Electronic J. Stat.*, **3**, 41–75.

Felzenszwalb, P. & Huttenlocher, D. (2006) Efficient belief propagation for early vision. *Int. J. Comput. Vis.*, **70**, 41–54.

Felzenszwalb, P. & Zabih, R. (2011) Dynamic programming and graph algorithms in computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, **33**, 721–740.

Fisher, N. (1995) *Statistical Analysis of Circular Data*. Cambridge: Cambridge University Press.

Fisher, N. & Lewis, T. (1983) Estimating the common mean direction of several circular or spherical distributions with differing dispersions. *Biometrika*, **70**, 333–341.

Fletcher, P. (2013) Geodesic regression and the theory of least squares on Riemannian manifolds. *Int. J. Comput. Vis.*, **105**, 171–185.

Fletcher, P. T., Venkatasubramanian, S. & Joshi, S. (2009) The geometric median on Riemannian manifolds with application to robust atlas estimation. *NeuroImage*, **45**, S143–S152.

Fornasier, M. & Ward, R. (2010) Iterative thresholding meets free-discontinuity problems. *Found. Comput. Math.*, **10**, 527–567.

Forney Jr, G. (1973) The Viterbi algorithm. *Proc. IEEE*, **61**, 268–278.

Frick, K., Marnitz, P. & Munk, A. (2012) Statistical multiresolution Dantzig estimation in imaging: fundamental concepts and algorithmic framework. *Electronic J. Stat.*, **6**, 231–268.

Frick, K., Munk, A. & Sieling, H. (2014a) Multiscale change point inference. *J. R. Stat. Soc. Series B Stat. Methodol.*, **76**, 495–580.

Frick, S., Hohage, T. & Munk, A. (2014b) Asymptotic laws for change point estimation in inverse regression. *Stat. Sin.*, **24**, 555–575.

Friedrich, F., Kempe, A., Liebscher, V. & Winkler, G. (2008) Complexity penalized M-estimation: fast computation. *J. Comput. Graph. Stat.*, **17**, 201–224.

Fu, H., Ng, M., Nikolova, M. & Barlow, J. (2006) Efficient minimization methods of mixed ℓ^1-ℓ^1 and ℓ^2-ℓ^1 norms for image restoration. *SIAM J. Sci. Comput.*, **27**, 89–97.

Geman, S. & Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, **6**, 721–741.

Giaquinta, M., Modica, G. & Souček, J. (1993) Variational problems for maps of bounded variation with values in S^1. *Calc. Var. Partial Differ. Equ.*, **1**, 87–121.

Hotz, T., Schutte, O. M., Sieling, H., Polupanow, T., Diederichsen, U., Steinem, C. & Munk, A. (2013) Idealizing ion channel recordings by a jump segmentation multiresolution filter. *IEEE Trans. NanoBiosci.*, **12**, 376–386.

Hupé, P., Stransky, N., Thiery, J., Radvanyi, F. & Barillot, E. (2004) Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, **20**, 3413–3422.

Ising, E. (1925) Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik A: Hadrons and Nuclei*, **31**, 253–258.

Jackson, B., Scargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L. & Tsai, T. T. (2005) An algorithm for optimal partitioning of data on an interval. *IEEE Signal Process. Lett.*, **12**, 105–108.

Jammalamadaka, S. & SenGupta, A. (2001) *Topics in Circular Statistics*. Singapore: World Scientific Publishing Company, 2001.

Joo, C., Balci, H., Ishitsuka, Y., Buranachai, C. & Ha, T. (2008) Advances in single-molecule fluorescence methods for molecular biology. *Annu. Rev. Biochem.*, **77**, 51–76.

Killick, R., Fearnhead, P. & Eckley, I. (2012) Optimal detection of changepoints with a linear computational cost. *J. Am. Stat. Assoc.*, **107**, 1590–1598.

Kolmogorov, V., Pock, T. & Rolinek, M. (2016) Total variation on a tree. *SIAM J Imaging Sci*. **9:2**, 605–636.

Lellmann, J., Strekalovskiy, E., Koetter, S. & Cremers, D. (2013) Total variation regularization for functions with values in a manifold. *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 3-6, 2013. pp. 2944–2951.

Li, H., Munk, A. & Sieling, H. (2016) FDR-contol in multiscale change-point segmentation. *Electron. J. Statist.*, **10**, 918–959.

Little, M. & Jones, N. (2011a) Generalized methods and solvers for noise removal from piecewise constant signals. I. Background theory. *Proc. R. Soc. A Math. Phys. Eng. Sci.*, **467**, 3088–3114.

Little, M. & Jones, N. (2011b) Generalized methods and solvers for noise removal from piecewise constant signals. II. New methods. *Proc. R. Soc. A Math. Phys. Eng. Sci.*, **467**, 3115–3140.

Mora, T., Yu, H., Sowa, Y. & Wingreen, N. (2009) Steps in the bacterial flagellar motor. *PLoS Comput. Biol.*, **5**, e1000540.

Mumford, D. & Shah, J. (1985) Boundary detection by minimizing functionals. *IEEE Conference on Computer Vision and Pattern Recognition*, **17**, 137–154. San Francisco, California, June 19–23, 1985.

Mumford, D. & Shah, J. (1989) Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.*, **42**, 577–685.

Oller, J. & Corcuera, J. (1995) Intrinsic analysis of statistical estimation. *Ann. Stat.*, **23**, 1562–1581.

Pennec, X. (2006) Intrinsic statistics on Riemannian manifolds: basic tools for geometric measurements. *J. Math. Imaging Vis.*, **25**, 127–154.

Potts, R. (1952) Some generalized order-disorder transformations. *Math. Proc. Cambridge Philos. Soc.*, **48**, 106–109.

Snijders, A. M., Nowak, N., Segraves, R., Blackwood, S., Brown, N., Conroy, J., Hamilton, G., Hindle, A. K., Huey, B., Kimura, K., Law, S., Myambo, K., Palmer, J., Ylstra, B., Yue, J. P., Gray, J. W., Jain, A. N. Pinkel, D., & Albertson, D. G. (2001) Assembly of microarrays for genome-wide measurement of DNA copy number by CGH. *Nat. Genetics*, **29**, 263–264.

Sowa, Y. & Berry, R. (2008) Bacterial flagellar motor. *Q. Rev. Biophys.*, **41**, 103–132.

Sowa, Y., Rowe, A., Leake, M., Yakushi, T., Homma, M., Ishijima, A. & Berry, R. (2005) Direct observation of steps in rotation of the bacterial flagellar motor. *Nature*, **437**, 916–919.

Storath, M., Weinmann, A. & Unser, M. (2016) Exact algorithms for L^1-TV regularization of real-valued or circle-valued signals. *SIAM J. Sci. Comput.*, **38**, A614–A630.

Unser, M. & Tafti, P. (2014) *An Introduction to Sparse Stochastic Processes*. Cambridge: Cambridge University Press.

Viterbi, A. (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory*, **13**, 260–269.

Weinmann, A. & Storath, M. (2015) Iterative Potts and Blake-Zisserman minimization for the recovery of functions with discontinuities from indirect measurements. *Proc. R. Soc. A*, **471**.

Weinmann, A., Demaret, L. & Storath, M. (2014) Total variation regularization for manifold-valued data. *SIAM J. Imaging Sci.*, **7**, 2226–2257.

Weinmann, A., Demaret, L. & Storath, M. (2016) Mumford-Shah and Potts regularization for manifold-valued data. *J. Math. Imaging Vis.*, **55**, 428–445.

Weinmann, A., Storath, M. & Demaret, L. (2015) The L^1-Potts functional for robust jump-sparse reconstruction. *SIAM J. Numer. Anal.*, **53**, 644–673.

Winkler, G. & Liebscher, V. (2002) Smoothers for discontinuous signals. *J. Nonparametric Stat.*, **14**, 203–222.

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

Wittich, O., Kempe, A., Winkler, G. & Liebscher, V. (2008) Complexity penalized least squares estimators: analytical results. *Mathematische Nachrichten*, **281**, 582–595.

Xia, Z. & Qiu, P. (2015) Jump information criterion for statistical inference in estimating discontinuous curves. *Biometrika*, **102**, 397–408.

Yao, Y.-C. (1984) Estimation of a noisy discrete-time step function: Bayes and empirical Bayes approaches. *Ann. Stat.*, **12**, 1434–1447.

Appendix. Proofs

Proof of Lemma 1. Let $z \in \mathbb{T} \setminus V$ be arbitrary, and let us denote the median-defining functional by $f(v) = \sum_{n=1}^{N} w_n\, d(v, y_n)$. The diameter passing through z cuts the unit circle into two hemispheres. We denote the counterclockwise hemisphere relative to z by S^+ and the other one by S^-. Let us denote the total weight of the elements in S^+ by W^+, i.e., $W^+ = \sum_{n:y_n \in S^+} w_n$, and analogously let $W^- = \sum_{n:y_n \in S^-} w_n$. Without loss of generality, we can assume that S^- is not 'heavier' than S^+, i.e., $W^- \leq W^+$. Let $z' \in V$ be the nearest neighbor of z in V on S^+. The distance between each element in S^+ and z' is by $d(z', z)$ smaller than its distance to z. Hence, replacing z by z', the functional value of f is decreased by $\sum_{n:y_n \in S^+} w_n d(z', z) = d(z', z) W^+$. With an analogous argument for the other hemisphere, f is increased at the same time by $d(z', z) W^-$. Thus, we have that

$$f(z') = f(z) + d(z', z) W^- - d(z', z) W^+ = f(z) + d(z', z)(W^- - W^+) \leq f(z),$$

where the inequality follows from $W^- \leq W^+$. So, for every $z \in \mathbb{T} \setminus V$, there is z' in V that has a smaller or equal functional value as z. This implies that there is a median in V. □

Proof of Theorem 2. Let $Q = \{q_1, \ldots, q_t\} \subset \{1, \ldots, N-1\}$ be a set of t jump indices, and let X_Q be the set of $x \in X^N$ that jump at the index indicated by Q; that is, $x_n \neq x_{n+1}$ for $n \in Q$ and $x_n = x_{n+1}$ otherwise. The minimizer of the Potts restricted to X_Q may be expressed as

$$\min_{x \in X_Q} P_\gamma(x) = \gamma |Q| + \min_{x \in X_Q} \sum_{n=1}^{N} w_n\, d(x_n, y_n).$$

Since the elements of X_Q are constant between two jumps we get that

$$\min_{x \in X_Q} \sum_{n=1}^{N} w_n\, d(x_n, y_n) = \min_{\mu \in X} \sum_{n=1}^{q_1} w_n\, d(\mu, y_n) + \cdots + \min_{\mu \in X} \sum_{n=q_t+1}^{N} w_n\, d(\mu, y_n).$$

Hence, $x \in X_Q$ minimizes $\min_{x \in X_Q} P_\gamma(x)$ if x equals a median between any two jumps. By Lemma 1, a median is contained in the set V. Considering all potential sets of jump indices (which are finitely-many) completes the proof. □

Proof of Theorem 3. It follows from Theorem 2 and its subsequent paragraph that the search space reduction is $\mathcal{O}(KN)$. The utilized variant of the Viterbi algorithm solves the reduced problem in $\mathcal{O}(KN)$, see Felzenszwalb & Huttenlocher (2006). Thus, the overall method is $\mathcal{O}(KN)$. □

Proof of Proposition 4. We have that

$$P(|\eta_n^N| \geq m) \leq C \sum_{|j| \geq m} p^{|j|} = C \left[\left(2 \frac{1}{1-p} - p \right) - \left(2 \frac{1-p^m}{1-p} - p \right) \right] \leq C' p^m$$

for a constant $C' > 0$. Now let m_0 be such that $C' p^{m_0} < 1$. The cardinality of the unique values, i.e., $|\mathrm{Val}(\eta^N)|$, can be estimated by $|\mathrm{Val}(\eta^N)| \leq 2 \max_n |\eta_n^N| + 1$, where $|\eta_n^N|$ is the ordinary absolute value of η_n^N. It follows for $m \geq m_0$ that

$$P(|\mathrm{Val}(\eta^N)| < m) \geq P(\max_n |\eta_n^N| < (m-1)/2) = \prod\nolimits_{n=1}^{N} P(|\eta_n^N| < (m-1)/2)$$
$$\geq \prod\nolimits_{n=1}^{N} (1 - C'p^{(m-1)/2}) = (1 - C'p^{(m-1)/2})^N \geq 1 - C'Np^{(m-1)/2}.$$

Then, letting $m = -\frac{4}{\log p} \log N$ yields

$$P\Big(|\mathrm{Val}(\eta^N)| < -\frac{4}{\log p} \log N\Big) \geq 1 - C'p^{-1/2}Np^{\frac{2\log N}{-\log p}}$$
$$= 1 - C'p^{-1/2}N \exp\left(\frac{2\log N}{-\log p} \log p\right) = 1 - C'p^{-1/2}N^{-1}.$$

Hence $P\big(|\mathrm{Val}(\eta^N)| \geq C'' \log N\big) \to 0$ as $N \to \infty$, with $C'' = -\frac{4}{\log p}$, which implies that $|\mathrm{Val}(\eta^N)| \in \mathcal{O}_p(\log N)$. □

Proof of Lemma 7. Since $\hat{x}$ is a minimizer of the Potts functional, it is a minimizer of the problem

$$\min_x \sum\nolimits_{n=1}^{N} w_n \; d(x_n, y_n), \quad \text{s.t. } \|\nabla x\|_0 = \|\nabla \hat{x}\|_0. \tag{A.1}$$

By the minimality of $\hat{x}$ with respect to the Potts functional, we further get that $\sum_{n=1}^{N} w_n \, d(x_n, y_n) > \sum_{n=1}^{N} w_n \, d(\hat{x}_n, y_n)$ for all x with $\|\nabla x\|_0 < \|\nabla \hat{x}\|_0$. This means that all candidates with less than or equal to many jumps as $\hat{x}$ do not lead to a smaller value of the target functional in (A.1). Therefore, $\hat{x}$ is a solution of ($\mathcal{P}_J$) for the parameter $J = \|\nabla \hat{x}\|_0$. □

Proof of Theorem 8. It follows from Lemma 7 that the set of solutions of the Potts problem ($\mathcal{P}_\gamma$) is contained in the set of solutions of the jump constrained problem ($\mathcal{P}_J$). The latter can be computed for all number of jumps in $\mathcal{O}(KNL)$ by (6). For mapping the parameter value γ to the smallest number of jumps of a corresponding Potts solution ($\mathcal{P}_\gamma$), we need to compute intersections of certain lines and graphs of concave functions as explained in the paragraph before Theorem 8. Namely, in step $J+1$ $(0 \leq J \leq L)$ we compute the intersection of the line g^{J+1} with the concave piecewise affine function ϕ^J. (Here, we use the notation as in the paragraph before Theorem 8.) The costs for this can be bounded from above by $\mathcal{O}(L^2)$, and, as $L \leq N-1$, also by $\mathcal{O}(NL)$. Summing all up, we get the complexity $\mathcal{O}(KNL)$. □