Mathematical Programming Algorithms for Regression-Based Nonlinear Filtering in $\mathbb{R}^N$

Nicholas D. Sidiropoulos, *Member, IEEE,* and Rasmus Bro

***Abstract*— This paper is concerned with regression under a "sum" of partial order constraints. Examples include locally monotonic, piecewise monotonic, runlength constrained, and unimodal and oligomodal regression. These are of interest not only in nonlinear filtering but also in density estimation and chromatographic analysis. It is shown that under a least absolute error criterion, these problems can be transformed into appropriate finite problems, which can then be efficiently solved via dynamic programming techniques. Although the result does not carry over to least squares regression, hybrid programming algorithms can be developed to solve least squares counterparts of certain problems in the class.**

***Index Terms*— Dynamic programming, locally monotonic, monotone regression, nonlinear filtering, oligomodal, piecewise monotonic, regression under order constraints, runlength constrained, unimodal.**

I. Introduction

FOR THE purposes of this paper, a (nonparametric) regression problem is an optimization problem of the following form: Given a vector $\mathbf{y} = \{y(n)\}_{n=1}^N \in \mathbb{R}^N$, find $\mathbf{x} = \{x(n)\}_{n=1}^N$ to

$$\begin{aligned} &\textbf{minimize: } d(\mathbf{y}-\mathbf{x}) \\ &\textbf{subject to: } \mathbf{x} \in \Phi \end{aligned}$$

where $d(\cdot)$ is typically some metric or semi-metric [1], and Φ is some set of feasible (or, *admissible*) solutions known as the *characteristic set* of the regression.

A common $d(\cdot)$ is the ℓ_2 norm (Euclidean distance to the origin), leading to least squares (LS) regression. LS regression is optimal (in the maximum likelihood sense) when measurement errors are additive, i.i.d. Gaussian. Gaussianity is an often-made assumption, for both practical (tractability) and theoretical (central limit theorem [2]) considerations. Another common $d(\cdot)$ is the ℓ_1 norm, leading to least absolute error (LAE) regression. The use of the ℓ_1 norm instead of the ℓ_2 norm may add a certain measure of robustness to the regression: LAE regression is optimal (in the maximum likelihood sense) when measurement errors are additive, i.i.d. Laplacian. The Laplacian is a much longer tailed distribution than the Gaussian; therefore, it is better suited to model impulsive noise. Constant LAE regression for $N = 2\nu + 1$ $(\nu \geq 1)$ samples can reject up to ν impulses (outliers), regardless of strength; constant LS regression may be significantly affected by such impulses. Refer to [3] for a discussion on the statistical optimality of regression under a semimetric.

Following [1], given any $\mathbf{x} \in \mathbb{R}^N$, we define its associated *sign skeleton* $\mathbf{s_x} \in \{-1, 0, 1\}^{N-1}$ as

$$s_{\mathbf{x}}(n) = \begin{cases} -1, & \text{if } x(n+1) < x(n) \\ +1, & \text{if } x(n+1) > x(n) \\ 0, & \text{if } x(n+1) = x(n). \end{cases}$$

Definition 1: A *skeletonizable* constraint set is any Φ such that membership of $\mathbf{x}$ in Φ can be determined by sole knowledge of its sign skeleton $\mathbf{s_x}$.

This paper deals with regression under skeletonizable constraint sets. Note that since membership of $\mathbf{x}$ in Φ can be determined by sole knowledge of its sign skeleton, there exists some $\Theta(\Phi) \subseteq \{-1, 0, 1\}^{N-1}$ such that $\mathbf{x} \in \Phi$ if and only if $\mathbf{s_x} \in \Theta(\Phi)$, in which case, $\min_{\mathbf{x}\in\Phi} d(\mathbf{y}-\mathbf{x}) = \min_{\mathbf{s_x}\in\Theta(\Phi)}[\min_{\mathbf{x}|\mathbf{s_x}} d(\mathbf{y}-\mathbf{x})]$. The inner minimization is a regression under a partial order constraint. LS regression under partial order constraints is the subject of [4]. The overall minimization is a regression over a union ("sum") of partial order constraints.

Another issue is whether or not Φ allows the elements of $\mathbf{x}$ to take on continuous or only discrete (and finitely many) values. Although the latter is often the case of interest in digital filtering applications [5]–[7], in many other applications (e.g., chromatography), we are interested in regression in $\mathbb{R}^N$ [8].

This paper deals with LAE and LS regression in $\mathbb{R}^N$ under skeletonizable constraint sets Φ. We focus on a selected subset of problems in this class, namely, locally monotonic, piecewise monotonic, runlength constrained, and unimodal and oligomodal regression. These are of interest in segmentation and nonlinear filtering [1], [3], [5], [6], [7], [9] density estimation [4], psychology [10], databases [11], biology [12], texture perception [13], and optimum decoding for magnetic media storage [14]. In the context of nonlinear filtering, these regressions offer optimal counterparts of standard nonlinear filters, such as iterated median or morphological filters, while obviating the restriction to monotone increasing operators [5], [6]. It is shown that under a least absolute error criterion, these problems can be transformed into appropriate *finite* problems, which can then be efficiently solved via dynamic programming (DP) techniques. Although the result does not carry over to

Manuscript received May 27, 1997; revised July 24, 1998. This work was supported by NSF ERC funds, the Martin-Marietta Chair in Systems Engineering, the Nordic Industry Foundation Project P93149, and the $F\emptyset TEK$ Foundation. The associate editor coordinating the review of this paper and approving it for publication was Dr. Peter C. Doerschuk.

N. D. Sidiropoulos is with the Department of Electrical Engineering, University of Virginia, Charlottesville, VA 22903 USA (e-mail: nikos@virginia.edu).

R. Bro is with the Chemometrics Group, Food Technology, The Royal Veternary and Agricultural University, Frederiksberg, Denmark (e-mail: rb@kvl.dk).

Publisher Item Identifier S 1053-587X(99)01341-0.

least squares regression, hybrid programming algorithms can be developed to solve least squares counterparts of certain problems in the class.

A. Organization

The rest of this paper is structured as follows. Section II presents a result that sets the stage for efficient DP solution of LAE regression under skeletonizable Φ. Section III is concerned with locally monotonic regression in $\mathbb{R}^N$ and includes background and motivation, a review of previous approaches, and some pertinent material on monotone regression and Kruskal's algorithm. Section IV is concerned with piecewise monotonic regression in $\mathbb{R}^N$. Section V presents simulation results for locally monotonic and piecewise monotonic regression. Section VI discusses an algorithm for runlength constrained regression, whereas Section VII discusses algorithms for unimodal and oligomodal regression. Conclusions are drawn in Section VIII.

II. A Quantization Result for Skeletonizable LAE Regressions

We have the following Lemma, which is a generalization of [1, Lemma 4].

Lemma 1: Suppose that $d(\cdot)$ is an ℓ_p norm, and $p \in [1, \infty)$, i.e., $d(\mathbf{y}-\mathbf{x}) = (\sum_{n=1}^{N} |y(n)-x(n)|^p)^{\frac{1}{p}}$ for some $p \in [1, \infty)$. Furthermore, suppose that Φ is skeletonizable. If $\mathbf{x}$ is a solution of

$$\begin{aligned} &\textbf{minimize: } d(\mathbf{y}-\mathbf{x}) \\ &\textbf{subject to: } \mathbf{x} \in \Phi \end{aligned}$$

then $\mathbf{x}$ is a piecewise-constant sequence whose pieces are constant regressions of the corresponding segments of $\mathbf{y}$ under the given distance metric.

Proof: Lemma 4 of [1] makes the same claim for the special case of locally monotonic regression. The proof follows along the lines of the proof in [1]. In particular, recall that since membership of $\mathbf{x}$ in Φ can be determined by sole knowledge of its sign skeleton $\mathbf{s_x}$, there exists some $\Theta(\Phi) \subseteq \{-1, 0, 1\}^{N-1}$ such that $\mathbf{x} \in \Phi$ if and only if $\mathbf{s_x} \in \Theta(\Phi)$, in which case, $\min_{\mathbf{x}\in\Phi} d(\mathbf{y}-\mathbf{x}) = \min_{\mathbf{s_x}\in\Theta(\Phi)}[\min_{\mathbf{x}|\mathbf{s_x}} d(\mathbf{y}-\mathbf{x})]$. Fix $\mathbf{s_x}$, and consider the inner minimization. An optimal $\mathbf{x}$ can always be thought of as a piecewise constant sequence (some or all of its pieces may contain just one element). If any given constant piece of $\mathbf{x}$ is not a constant regression of the corresponding elements of $\mathbf{y}$, then its fit (and, thus, the fit of $\mathbf{x}$) may always be improved by perturbing its level by an infinitesimal amount to bring it closer to the said regression *without changing the sign skeleton* $\mathbf{s_x}$. This is true because strict inequality allows for an open ball of free movement of level in either direction. This contradicts conditional optimality of $\mathbf{x}$ conditioned on $\mathbf{s_x}$. Thus, an optimal $\mathbf{x}$ for any given $\mathbf{s_x}$ is a piecewise-constant sequence whose pieces are constant regressions of the corresponding segments of $\mathbf{y}$ under the given distance metric. This holds for all $\mathbf{s_x}$, and the result follows. □

Lemma 2: If $d(\cdot)$ is ℓ_1 norm, i.e., $d(\mathbf{y}-\mathbf{x}) = \sum_{n=1}^{N} |y(n)-x(n)|$, and Φ is skeletonizable, then there exists a solution $\mathbf{x}_o$ of

$$\begin{aligned} &\textbf{minimize: } d(\mathbf{y}-\mathbf{x}) \\ &\textbf{subject to: } \mathbf{x} \in \Phi \end{aligned}$$

such that for all $n = 1, 2, \ldots, N$, there exists an $m \in \{1, 2, \ldots, N\}$ such that $x_o(n) = y(m)$.

Proof: Constant regression under ℓ_1 amounts to picking a median of the elements involved (e.g., [1]). If the number of elements L is odd, then the median is the $(\frac{L+1}{2})$-order statistic of the given elements; if L is even, then the closed interval delimited by the $(\frac{L}{2})$-order statistic and the $(\frac{L}{2}+1)$-order statistic is the set of all medians of the given elements [1]. Either way, one may always select a median from the given elements. □

Corollary 1: Suppose that $d(\cdot)$ is ℓ_1 norm. Define $\mathcal{A} = \{v \in \mathbb{R} \mid \exists n \in \{1, 2, \ldots, N\} : y(n) = v\}$. Furthermore, suppose that Φ is skeletonizable. Then, there exists a solution $\mathbf{x}_o$ of

$$\begin{aligned} &\textbf{minimize: } d(\mathbf{y}-\mathbf{x}) \\ &\textbf{subject to: } \mathbf{x} \in \Phi \end{aligned}$$

which is also a solution of

$$\begin{aligned} &\textbf{minimize: } d(\mathbf{y}-\mathbf{x}) \\ &\textbf{subject to: } \mathbf{x} \in \Phi \cap \mathcal{A}^N. \end{aligned}$$

All other solutions of the former problem are equivalent to $\mathbf{x}_o$ in the sense that they all achieve the same $d(\mathbf{y}-\mathbf{x})$. Any multiple solutions of the latter problem solve the former problem as well.

This is important because *it reduces regression over a subset of* $\mathbb{R}^N$ *to a finite problem*. In all cases of interest to us, Φ does not have the algebraic structure of a subspace, nor is it convex; thus, the original problem is difficult. The finite problem, on the other hand, often admits an efficient algorithmic solution.

In the sequel, we explore several regression problems that satisfy the conditions of this result. As we will see, the corresponding LS versions do not admit such a "universal" solution and have to be addressed on a one-by-one basis using different means.[1]

III. Locally Monotonic Regression in $\mathbb{R}^N$

A. Background on Locally Monotonic Regression

Locally monotonic regression is the optimal counterpart of iterated median filtering. In [1], Restrepo and Bovik developed an elegant mathematical framework in which they studied locally monotonic regressions in $\mathbb{R}^N$. They proved existence of such regressions and provided algorithms for their computation. However, the complexity of their algorithms is exponential in N, which is the size of the input sample. In addition to existence, Restrepo and Bovik showed that locally monotonic regression admits a maximum likelihood interpretation [3].

[1]The ℓ_1 norm is "special" in a particular sense. Other such strong results exist for the ℓ_1 norm but do not extend to the case of the ℓ_2 norm. A recent example appears in [15], where it is shown that a similar property holds for the solution of a particular *regularization* problem defined via ℓ_1 norms. Note that here, we instead have a *class* of *regression* problems for which our result is valid.

In [6], the first author considered *digital* locally monotonic regressions in which the output symbols are drawn from a finite alphabet and, by making a connection to Viterbi decoding, provided a fast (linear in N) algorithm that computes any such regression, be it under a metric, semi-metric, or arbitrary bounded per-letter cost measure.

A few definitions are in order. If $\mathbf{x}$ is a real-valued sequence (string) of length N and γ is any integer less than or equal to N, then a *segment* of $\mathbf{x}$ of length γ is any substring of γ consecutive components of $\mathbf{x}$. Let $\mathbf{x}_i^{i+\gamma-1} = \{x(i), \ldots, x(i+\gamma-1)\}$, $i \geq 0$, $i+\gamma \leq N$ be any such segment. $\mathbf{x}_i^{i+\gamma-1}$ is monotonic if either $x(i) \leq x(i+1) \leq \cdots \leq x(i+\gamma-1)$ or $x(i) \geq x(i+1) \geq \cdots \geq x(i+\gamma-1)$.

Definition 2: A real-valued sequence $\mathbf{x}$ of length N is *locally monotonic* of degree $\alpha \leq N$ (or *lomo*-α or simply *lomo* in case α is understood) if each and every one of its segments of length α is monotonic.

Notice that some segments may be increasing, others decreasing, and the sequence may still pass the test of local monotonicity. In general, monotonicity implies local monotonicity, but not vice versa. Any sequence is locally monotonic of degree $\alpha = 2$; therefore, the interesting degrees are 3 through N. Throughout the following, we assume that $3 \leq \alpha \leq N$. If $\alpha \leq \beta \leq N$, then a sequence of length N that is lomo-β is lomo-α as well; thus, the *lomotonicity* of a sequence is defined as the highest degree of local monotonicity that it possesses [1].

A sequence $\mathbf{x}$ is said to exhibit an increasing (resp. decreasing) transition at coordinate i if $x(i) < x(i+1)$ (resp. $x(i) > x(i+1)$). The following (cf. [1], [16], [17]) is a key property of locally monotonic signals: If $\mathbf{x}$ is locally monotonic of degree α, then $\mathbf{x}$ has a constant segment (run of identical elements) of length at least $\alpha - 1$ in between an increasing and a decreasing transition. The reverse is also true.

The study of local monotonicity (which led to the idea of locally monotonic regression) has a relatively long history in the field of nonlinear filtering. Local monotonicity appeared in the study of the set of root signals of the median filter [16]–[23]. The median is arguably the most widely known and used nonlinear filter. Locally monotonic signals of degree α are roots of the $[2(\alpha-1)-1]$-point 1-D median filter (meaning that they are not affected by filtering using a $[2(\alpha-1)-1]$-point median filter). Not all median roots are locally monotonic [23]; however, locally monotonic signals can be thought of as the class of "well-behaved" median roots, at least from a nonlinear filtering perspective. In practice, iterations of the median usually (but not always) converge to a locally monotonic approximation of the input signal. It is then natural to ask for the *best* possible locally monotonic approximation of the input at hand, rather than settle for the arbitrarily chosen locally monotonic approximation provided by iterated median filtering. This gives rise to locally monotonic (*lomo*) regression [1], [3]

$$\begin{aligned} &\textbf{minimize: } d(\mathbf{y}-\mathbf{x}) \\ &\textbf{subject to: } \mathbf{x} \in \Lambda_\alpha^N \end{aligned}$$

where Λ_α^N is the set of all sequences of N elements of $\mathbb{R}$ that are locally monotonic of lomo-degree α.

B. Previous Approaches

Previous algorithms for lomo regression include the following.

- The *Tube* and *Blotching* algorithms of Restrepo and Bovik [1]: These are of exponential complexity.
- The work of Sidiropoulos on fast digital locally monotonic regression [6]: This algorithm has complexity linear in the number of samples but solves a discretized problem.
- The recent work of de la Vega and Restrepo [24], in which the authors show that, in computing a LS lomo-α regression, we do not need flat segments of length greater than or equal to $2(\alpha-1)$: This result can be utilized to cut down search complexity and construct more efficient LS lomo regression algorithms. The authors note that complexity is polynomial in the size of the sample.

In the sequel, we present an efficient exact algorithm for locally monotonic LAE regression in $\mathbb{R}^N$ and a fast algorithm for locally monotonic LS pseudo-regression in $\mathbb{R}^N$.

C. Locally Monotonic LAE Regression in $\mathbb{R}^N$

Consider the locally monotonic LAE regression problem

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y}-\mathbf{x}\|_1 \\ &\textbf{subject to: } \mathbf{x} \in \Lambda_\alpha^N. \end{aligned}$$

It is easy to see that Λ_α^N satisfies the condition of Corollary 1. Define $\mathcal{A}$ as in Corollary 1, and consider the resulting digital problem

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y}-\mathbf{x}\|_1 \\ &\textbf{subject to: } \mathbf{x} \in \Lambda_\alpha^N \cap \mathcal{A}^N. \end{aligned}$$

This is an instance of digital locally monotonic regression, which has been solved in [6] by means of DP. The complexity of the digital algorithm is $O(|\mathcal{A}|^2 \alpha N)$. $\mathcal{A}$ has, at most, N elements. It follows that the computational complexity of locally monotonic LAE regression in $\mathbb{R}^N$ is $O(N^3\alpha)$. This solution is *exact* and *efficient*, although it may be quite tedious for long observation sequences.

We may improve on complexity by capitalizing on the following observation. Since, at any given point, the value of the optimum solution is the median of a small number of input samples in the neighborhood of the given point, we may restrict the alphabet at time n to be the set of input values in a suitable neighborhood about n. The longest streaks in a locally monotonic regression of degree α are usually bounded in length by $k \times \alpha$, where k is a small integer constant; we may therefore pick a *local alphabet* for $x(n)$ that consists of, say, at most 4α elements and run DP under this restriction. This is beneficial for α small relative to N and has complexity $O(\alpha^3 N)$.

D. Monotone Regression

Monotone (say, increasing) LS regression has the following problem: Given $\mathbf{y}$, find $\mathbf{x}$ to

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y}-\mathbf{x}\|_2^2 \\ &\textbf{subject to: } \mathbf{x} \textit{ is monotone increasing} \end{aligned}$$

or, to be more precise, monotone nondecreasing $x(1) \leq \cdots \leq x(N)$. This is a special case of *isotonic* regression (with respect to a simple order) [4]. In 1964, Kruskal [10] came up with what appears to be the best way of solving this problem, which he studied in the context of his pioneering work in *multidimensional scaling*; see also Leeuw [25], Barlow *et al.* [4], Cran [26], and Bril *et al.* [27]. The computational algorithm is often referred to as the *up and down blocks (UDB) algorithm* [4]. Kruskal's approach is *iterative* and makes use of a simple but clever observation to reduce the problem of monotone LS regression to a finite sequence of averaging steps (this can be understood within the framework of Lemma 1). The theoretical complexity of Kruskal's algorithm is (rather loosely) upper bounded by $O(N^2)$; in practice, it is almost $O(N)$.

For our purposes, as we will soon see, we will need a fast algorithm for a slightly different problem. In particular, we will need an algorithm for the *suffix-constrained nondecreasing regression* problem

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y}-\mathbf{x}\|_2^2 \\ &\textbf{subject to: } x(1) \leq \cdots \leq x(N-\alpha+2) = \cdots = x(N) \end{aligned}$$

i.e., the last $\alpha - 1$ elements of the regression should be equal, for some $3 \leq \alpha \leq N$, as well as the corresponding nonincreasing problem

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y}-\mathbf{x}\|_2^2 \\ &\textbf{subject to: } x(1) \geq \cdots \geq x(N-\alpha+2) = \cdots = x(N). \end{aligned}$$

Suppose that we have an algorithm that solves the *prefix-constrained nondecreasing regression* problem

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y}-\mathbf{x}\|_2^2 \\ &\textbf{subject to: } x(1) = x(2) = x(\alpha-1) \leq x(\alpha) \leq \cdots \leq x(N). \end{aligned}$$

Then, it is easy to see that the first of the two problems of interest can be solved by feeding $-\text{rev}(\mathbf{y})$ (where the $\text{rev}(\cdot)$ operation simply reverses the order of elements of its argument) as input to the said algorithm and then computing $-\text{rev}(\mathbf{x})$, whereas the second problem of interest can be solved by feeding $\text{rev}(\mathbf{y})$ to the said algorithm and then computing $\text{rev}(\mathbf{x})$. These statements can be verified using only that $\|\mathbf{y}-\mathbf{x}\|_2^2 = \|\text{rev}(\mathbf{y}) - \text{rev}(\mathbf{x})\|_2^2$, and $\|\mathbf{x}\|_2^2 = \|-\mathbf{x}\|_2^2$.

It will be beneficial to be able to compute all subregressions, i.e., in the context of the prefix-constrained nondecreasing regression algorithm predicated above, it will be useful to be able to obtain as byproducts all prefix-constrained subregressions on the first k elements of $\mathbf{y}$ for $k \geq \alpha - 1$. Given Kruskal's basic monotone regression algorithm [10], the construction of an algorithm that incorporates the desired additional features is a simple exercise. The details are left out due to space considerations, but MATLAB code is available at http://www.people.virginia.edu-nds5j.

E. Locally Monotonic LS Regression in $\mathbb{R}^N$

Let us now consider locally monotonic LS regression

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y}-\mathbf{x}\|_2^2 \\ &\textbf{subject to: } \mathbf{x} \in \Lambda_\alpha^N. \end{aligned}$$

This problem turns out to be more difficult than its LAE counterpart. In particular, we cannot resort to a result similar to Corollary 1: The set of all partial averages over a given set of reals can be very large.

A suboptimum but efficient way to proceed is as follows. By proper discretization, digital locally monotonic regression can provide an approximate solution to the underlying regression in $\mathbb{R}^N$. This interim solution can be improved by applying the suffix-constrained monotone regression algorithm in between adjacent breakpoints of the interim solution. If the resulting regression respects local endpoint consistency constraints (e.g., that the first element of an increasing segment following a decreasing segment should be greater than the last element of the decreasing segment), then inserting it in place of the corresponding piece of the interim solution will improve the fit without violating local monotonicity constraints.

Since the digital algorithm is linear in N (but quadratic in the size of the digital alphabet—meaning that very fine discretization can be costly) and the suffix-constrained monotone regression algorithm is better than quadratic in N, it follows that the complexity of this suboptimum two-step process is loosely bounded above by $O(N^2)$ and much better than this figure in practice. In Section V, we show that this two-step approach provides measurable improvements in fit relative to the digital algorithm, at a small extra complexity cost.

IV. Piecewise Monotonic LS Regression in $\mathbb{R}^N$

Piecewise monotonic (*pimo*) LS regression of degree α has the following problem:

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y}-\mathbf{x}\|_2^2 \\ &\textbf{subject to: } \mathbf{x} \in \Pi_\alpha^N \end{aligned}$$

where Π_α^N is the set of all sequences of N elements of $\mathbb{R}$ that can be constructed by concatenating monotone pieces, each one of which can be either increasing or decreasing *and* of length at least $\alpha - 1$. If $\mathbf{x} \in \Pi_\alpha^N$, then we say that $\mathbf{x}$ is piecewise monotonic of pimo-degree α. Note that $\beta \geq \alpha \Rightarrow \Pi_\beta^N \subseteq \Pi_\alpha^N$; the minimum is a nondecreasing function of α.

There are several reasons for introducing piecewise monotonic regression. From an optimization viewpoint, piecewise monotonic regression relieves some of the constraints associated with local monotonicity. In particular, it relieves the segment endpoint consistency constraints (e.g., that the first element of an increasing segment following a decreasing segment should be greater than the last element of the decreasing segment) as well as the constraint that individual monotone segments should terminate with a suffix string of at least $\alpha - 1$ identical samples. The latter generates undesirable *streaking* effects that are similar to median filtering [28]. The idea is that by relaxing the suffix constraint, we may be able to reduce[2] streaking without sacrificing noise suppression and edge-preservation capabilities of the regression. This is indeed the case, as we will soon see in Section V by means of simulation. The fact that relaxing some of the local monotonicity constraints effectively *decouples* the problem in a particular sense and allows us to develop a fast hybrid DP-Kruskal algorithm for piecewise monotonic LS regression is also

[2] Streaking can be also attributed in part to the use of an ℓ_p norm $p = 1, 2$ and cannot be avoided altogether. This is implicit in Lemma 1.

important. Note that Π_α^N satisfies the condition of Corollary 1 and that the digital counterpart of this problem admits efficient solution via DP in a way similar to [6]. Therefore, LAE pimo regression in $\mathbb{R}^N$ can be performed at a complexity cost of $O(N^3\alpha)$.

A. Hybrid Programming Algorithm for Piecewise Monotonic LS Regression

Given $\mathbf{x} \in \Pi_\alpha^N$, we define its associated *trend switching set* $T(\mathbf{x})$ to be the set of indices $\{t_1, t_2, \ldots, t_K\}$, where $\mathbf{x}$ exhibits a trend transition (increasing to decreasing or vice versa).[3] Let us assume, without loss of generality, that $\mathbf{x}$ starts with an increasing trend.[4]

Convention 1: By convention, we allow the elements of $T(\mathbf{x})$ to take on the value N (which serves as a "gate" for unused trend switches); this allows us to fix $K := \text{card}[T(\mathbf{x})] = \text{floor}(\frac{N}{\alpha-1})$.

It is straightforward to verify the following proposition.

Proposition 1: Under Convention 1, $\mathbf{x} \in \Pi_\alpha^N$ if and only if $T(\mathbf{x})$ satisfies the following three properties:

Property 1: $\alpha - 1 \leq t_1 \leq t_2 \leq \cdots \leq t_K \leq N$.

Property 2: If $t_i \neq N$, then $t_{i+1} - t_i \geq \alpha - 1$, $i = 1, 2, \ldots, K-1$.

Property 3: If $t_i = N$, then $t_j = N$, $\forall K \geq j > i$.

Remark 1: In between $[t_{i-1}+1, t_i]$, $\mathbf{x}$ is constrained to be monotonic (specifically, nondecreasing for i: odd; nonincreasing for i: even); otherwise, it is unrestricted.

Let us denote by V_α^N the set of all positive integer K-tuples that satisfy Properties 1, 2, and 3. By the above proposition, $\mathbf{x} \in \Pi_\alpha^N$ if and only if $T(\mathbf{x}) \in V_\alpha^N$. Therefore, the piecewise monotonic least squares regression problem

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y} - \mathbf{x}\|_2^2 \\ &\textbf{subject to: } \mathbf{x} \in \Pi_\alpha^N \end{aligned}$$

is equivalent to

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y} - \mathbf{x}\|_2^2 \\ &\textbf{subject to: } T(\mathbf{x}) \in V_\alpha^N. \end{aligned}$$

Define $\mathcal{J}(\mathbf{x}) = ||\mathbf{y}-\mathbf{x}||_2^2$; then, $\min_{\mathbf{x}|T(\mathbf{x})\in V_\alpha^N} \mathcal{J}(\mathbf{x})$is equal to

$$\begin{aligned} &\min_{\{t_1,t_2,\ldots,t_K\}\in V_\alpha^N} \{\min \mathcal{J}(\mathbf{x} \mid T(\mathbf{x}) = \{t_1, t_2, \ldots, t_K\})\} \\ &= \min_{\{t_1,t_2,\ldots,t_K\}\in V_\alpha^N} \sum_{i=1}^{K} \min \mathcal{E}(t_{i-1}, t_i) \\ &= \min_{\{t_1,t_2,\ldots,t_K\}\in V_\alpha^N} \sum_{i=1}^{K} \mathcal{E}^*(t_{i-1}, t_i) \end{aligned}$$

[3]We may determine the trend switching set by knowledge of the sign skeleton but not vice versa. In the case of piecewise monotonic regression, feasibility of a candidate solution may be determined either from its sign skeleton or its trend switching set. The latter leads to a more compact parameterization of the feasible set.

[4]Otherwise, we simply run the process twice—one time starting with an increasing trend and the other with a decreasing trend—and pick the best of the two. This doubles the amount of computation but does not affect complexity order. Alternatively, we may allow the first monotone regression (and *only* the first) to be of zero length; this accomplishes the same goal at a smaller computational cost.

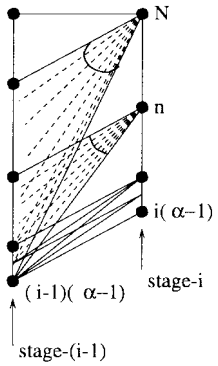


Fig. 1. Nodes (states) at stage i and predecessors at stage $(i-1)$. At stage i, the bottom node has just one predecessor that is α nodes apart. The next node going upwards has just two predecessors, the furthest of which is $\alpha+1$ nodes apart. A generic node n at stage i has $n - i(\alpha-1)+1$ predecessors, the furthest of which is $n-(i-1)(\alpha-1)+1$ apart. The top node (N) is $N-(i-1)(\alpha-1)+1$ apart from its furthest predecessor. Note that node N is special in the sense that it is possible to move from node N at stage $(i-1)$ to node N at stage i at zero cost.

where $t_0 = 0$ by convention, and $\mathcal{E}^*(t_{i-1}, t_i)$ is the cost of monotone (specifically, nondecreasing for i: odd; nonincreasing for i: even) least squares regression on the elements of $\mathbf{y}$ between indices $t_{i-1}+1$ and t_i inclusive.

The above derivation is related in spirit to an algorithm of Blake for a nonlinear regularization problem [29] and an algorithm of Bellman for a piecewise linear approximation problem subject to a budget on the number of pieces [30]. The key to all is dynamic programming over "breakpoint" variables. The Viterbi algorithm (VA) is the most familiar example of dynamic programming. In a nutshell, the VA is nothing but a clever method to search for an N-tuple $\{s(n)\}_{n=0}^{N-1}$ of finite-alphabet "state" variables that minimizes $\sum_{n=0}^{N-1} c_n(s(n), s(n-1))$, where $s(-1)$ is given, and $c_n(\cdot,\cdot)$ is some arbitrary "one-step transition" cost. The VA avoids exhaustive search and makes it possible to find an optimum solution in time linear in N (the proportionality constant depends on the cardinality of the finite alphabet and the specific cost structure) [31]–[34]. The problem at hand is in a form suitable for dynamic programming over the trend switching variables.

Consider Fig. 1, which depicts nodes (states) at stage i and their respective potential predecessors at stage $(i-1)$. At stage i, the node tags correspond to all the possible values of t_i and similarly for stage $(i-1)$. At stage i, the bottom node has just one predecessor that is α nodes apart. The next node going upwards has just two predecessors, the furthest of which is $\alpha+1$ nodes apart. Node n at stage i has $n-i(\alpha-1)+1$ predecessors, the furthest of which is $n-(i-1)(\alpha-1)+1$ apart. The top node (N) is $N-(i-1)(\alpha-1)+1$ apart from its furthest predecessor.

Each node at stage i is visited in turn, and a decision is made as to which of the associated potential predecessors is best for the node at hand. To do this, we need to calculate $\mathcal{E}^*(t_{i-1}, t_i)$ for the specific value of t_i assigned to the node at hand and *all* values of t_{i-1} assigned to its potential predecessor nodes; add the respective results to the corresponding cumulative costs of the potential predecessor nodes; pick the one that gives

minimum error; update the cumulative cost of the node at hand; set up a pointer to its best predecessor, and then move on to the next node, the next stage, and so on.

The important observation is that a "generic" node, say, the one that has been assigned the value $t_i = n$, only needs to make *one* call to Kruskal's monotone regression algorithm, and this call suffices to compute the node transition costs $\mathcal{E}^*(t_{i-1}, n)$ for all potential predecessors. Indeed, according to our earlier discussion, node n only needs to call the algorithm once with the longest input, and all required subregressions will be computed along the way for free. The length of this longest input is $n - (i-1)(\alpha-1) + 1$.

Since Kruskal's algorithm is better than quadratic, it follows that the computational cost for all required computations for the worst (top) node at stage i is bounded above by $O([N-(i-1)(\alpha-1)+1]^2)$. Stage i has a total of $[N-i(\alpha-1)+1]$ nodes; thus, the computational cost for all required computations for stage i is (quite loosely) bounded above by

$$O([N - i(\alpha-1) + 1] \times [N - (i-1)(\alpha-1) + 1]^2).$$

There exist at most $\frac{N}{\alpha-1}$ stages, and the worst stage is $i = 1$; therefore, the total computational cost for the entire regression is bounded above by

$$O\left(\frac{N}{\alpha-1}[N - (\alpha-1) + 1](N+1)^2\right).$$

This bound is rather conservative, but it seems hard to improve for arbitrary α (more sophisticated counting arguments result in a bound of the same order).

V. Simulation

The purpose of this section is to enhance the reader's intuition by providing and discussing the results of several simulation experiments on locally monotonic and piecewise monotonic LS regression in $\mathbb{R}^N$.

A. Locally Monotonic LS Regression in $\mathbb{R}^N$

We now investigate the fit versus speed tradeoff between the two-step algorithm and the digital algorithm for locally monotonic LS pseudo-regression. We shall see that for a small additional complexity cost, the two-step algorithm can significantly improve fit.

The two efficient algorithms have been implemented in `C` and `MATLAB`. Figs. 2–13 present the results of several experiments. Numerical results are summarized in Table I. The input $\mathbf{y}$ is a 300-point $(N = 300)$ sample of a noisy version of an ECG signal taken from the signal processing information base (http://www.spib.rice.edu). In all figures, the spike train at the bottom depicts trend switches, as detected by the digital algorithm. In the captions, $|\mathcal{A}|$ is the size of the digital alphabet, and α is the lomo-degree.

Figs. 2–4 present the results for $|\mathcal{A}| = 40$, $\alpha = 20$. In particular, Fig. 2 presents the input versus the two-step real-valued regression for $|\mathcal{A}| = 40$, $\alpha = 20$. Fig. 3 presents the input versus the digital regression for the same $|\mathcal{A}|$, α. Fig. 4 compares the two-step real-valued regression and the

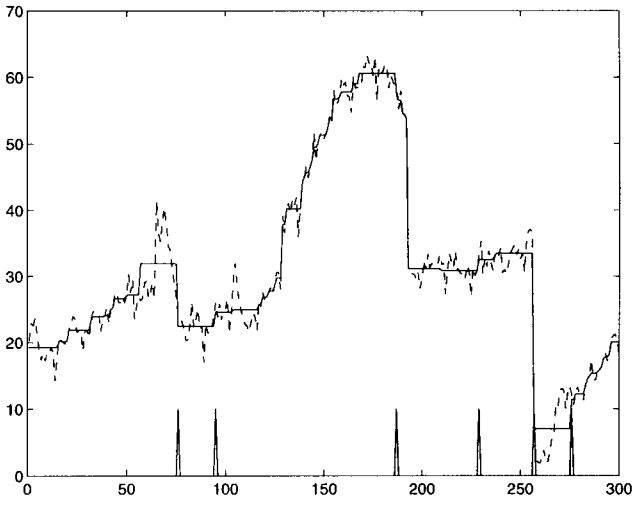


Fig. 2. Input (dashed) versus two-step real-valued (solid) regression $|\mathcal{A}| = 40$, $\alpha = 20$.

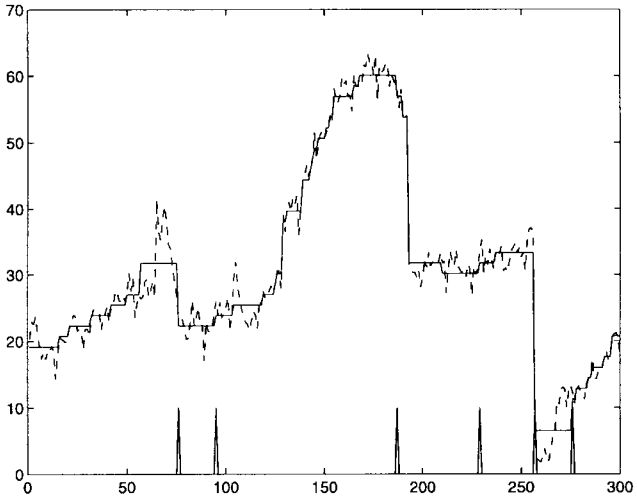


Fig. 3. Input (dashed) versus digital (solid) regression $|\mathcal{A}| = 40$, $\alpha = 20$.

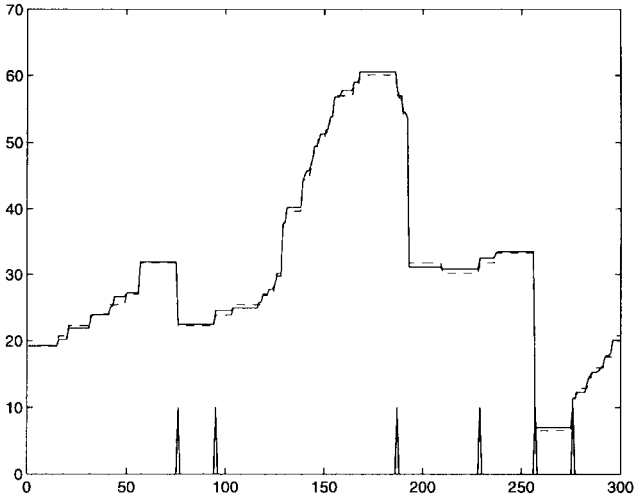


Fig. 4. Two-step real-valued (solid) versus digital (dashed) regression $|\mathcal{A}| = 40$, $\alpha = 20$.

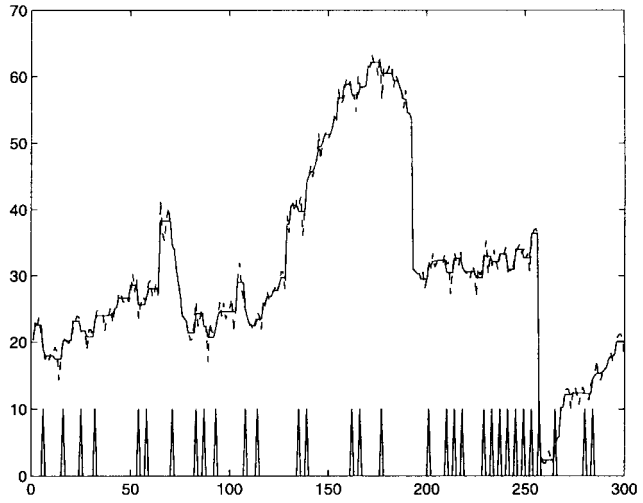


Fig. 5. Input (dashed) versus two-step real-valued (solid) regression $|\mathcal{A}| = 40,\ \alpha = 5$.

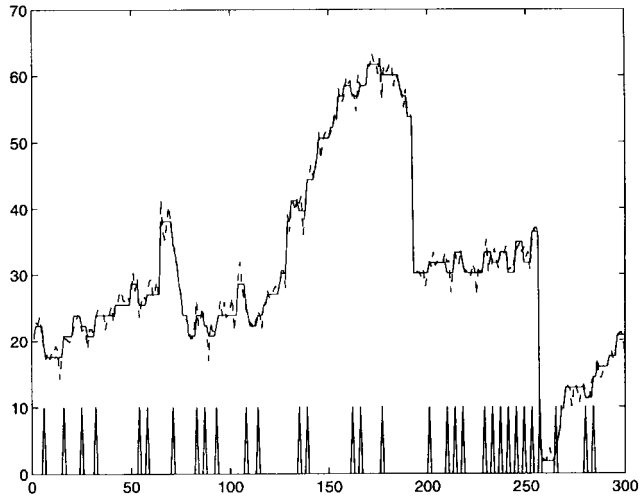


Fig. 6. Input (dashed) versus digital (solid) regression, $|\mathcal{A}| = 40,\ \alpha = 5$.

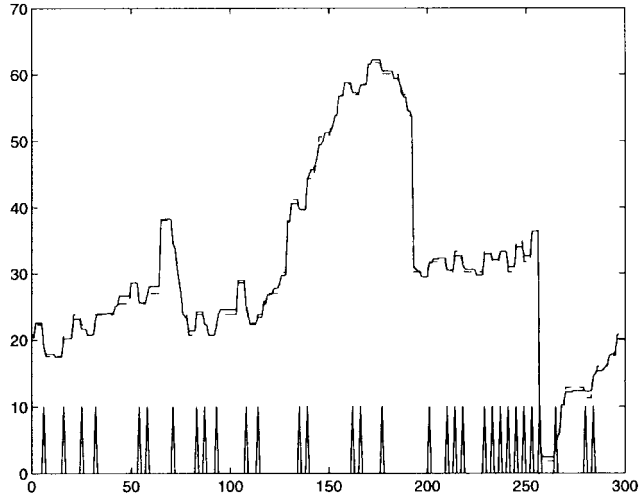


Fig. 7. Two-step real-valued (solid) versus digital (dashed) regression $|\mathcal{A}| = 40,\ \alpha = 5$.

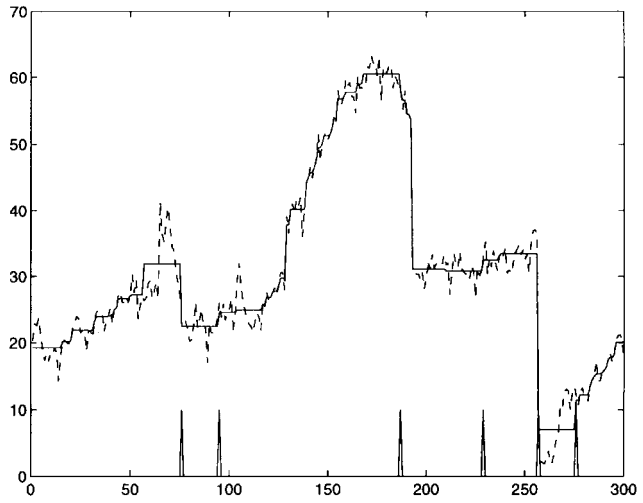


Fig. 8. Input (dashed) versus two-step real-valued (solid) regression $|\mathcal{A}| = 20,\ \alpha = 20$.

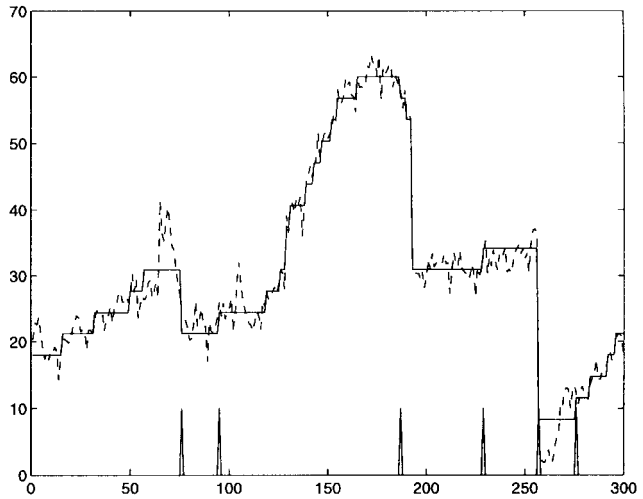


Fig. 9. Input (dashed) versus digital (solid) regression, $|\mathcal{A}| = 20,\ \alpha = 20$.

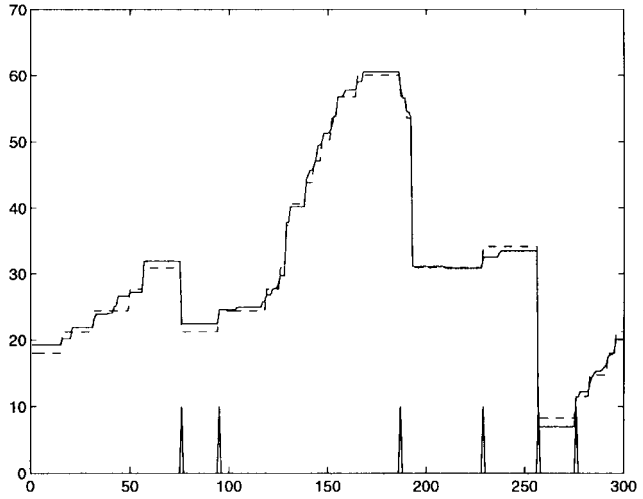


Fig. 10. Two-step real-valued (solid) versus digital (dashed) regression, $|\mathcal{A}| = 20,\ \alpha = 20$.

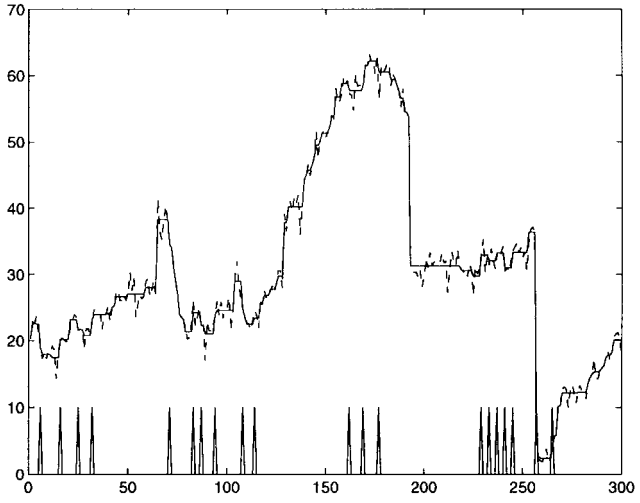


Fig. 11. Input (dashed) versus two-step real-valued (solid) regression, $|\mathcal{A}| = 20,\ \alpha = 5$.

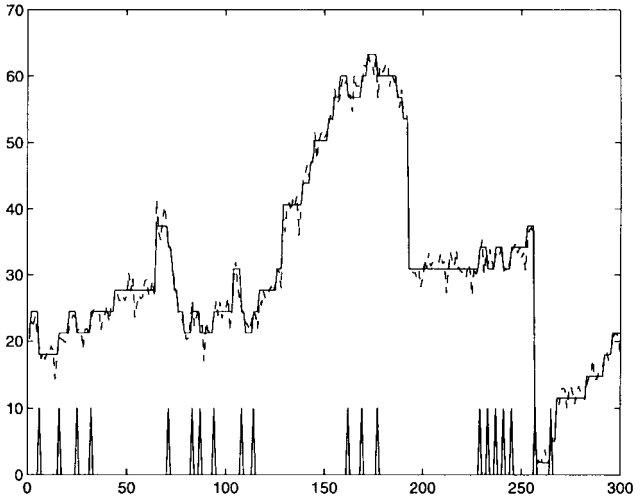


Fig. 12. Input (dashed) versus digital (solid) regression, $|\mathcal{A}| = 20,\ \alpha = 5$.

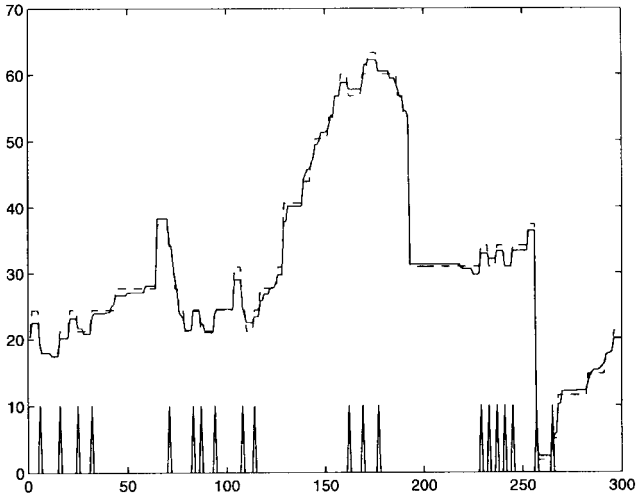


Fig. 13. Two-step real-valued (solid) versus digital (dashed) regression, $|\mathcal{A}| = 20,\ \alpha = 5$.

TABLE I
FIT AND CPU TIME (IN MINUTES'SECONDS"-C AND MATLAB ON SUN SPARC10, $N = 300$) FOR LOMO LS REGRESSION. f_D (f_{2S}): FIT, DIGITAL (TWO-STEP); T_D (T_{2S}): CPU TIME, DIGITAL (TWO-STEP)

$\lvert\mathcal{A}\rvert,\ \alpha$	f_D	f_{2S}	T_D	T_{2S}
40, 50	6946	6910	1'45"	1'47"
40, 20	1593	1518	40"	41"
40, 5	447	358	15"	15.5"
20, 50	6998	6912	30"	32"
20, 20	1775	1518	15"	16"
20, 5	640	417	5"	5.7"

corresponding digital regression. Subsequent triples of figures (through Fig. 13) do the same for various values of $|\mathcal{A}|$, α.

Let us consider the numerical results summarized in Table I. An important general observation is that the difference in run time between the digital and the two-step real-valued algorithm is negligible. In absolute terms, both run in the order of a few seconds, or a couple of minutes, at worst. In terms of fit, the two-step algorithm always improves on the fit of the digital algorithm, sometimes by as much as 30%. This improvement becomes more pronounced as $|\mathcal{A}|$ is reduced (which also leads to a rapid decrease in complexity since the digital algorithm is quadratic in $|\mathcal{A}|$). Furthermore, the percent improvement in fit afforded by the two-step algorithm increases with decreasing α. This may be intuitively explained as follows. When α is small, the digital algorithm is forced to closely track the input, thereby exhibiting tracking oscillations in a manner similar to a phenomenon known as *slope over/under load* in delta modulation [35].

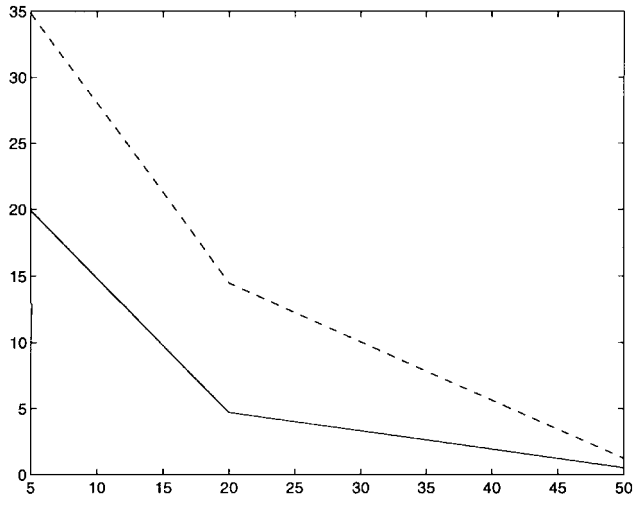
From Table I, we may also observe that the second step of the two-step algorithm is able to compensate for inaccuracies caused by using a small alphabet in the first (digital) step. Note that the fit of the two-step algorithm remains essentially the same, regardless of whether $|\mathcal{A}|$ is 40 or 20, at least for $\alpha = 50$, $\alpha = 20$, with a small degradation when $\alpha = 5$.

Fig. 14(a) plots the percent improvement in LS fit afforded by the two-step algorithm versus the digital algorithm, whereas Fig. 14(b) plots CPU time for the two-step algorithm as a function of α and $|\mathcal{A}|$.

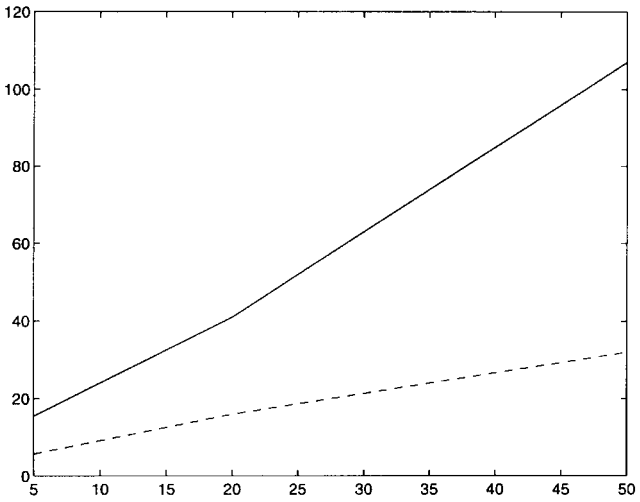
B. Piecewise Monotonic LS Regression in $\mathbb{R}^N$

The results of experiments on piecewise monotonic LS regression are presented in Fig. 15 and Table II. Fig. 15 presents the result of piecewise monotonic regression of pimo-degree $\alpha = 50, 20$, respectively, for the same noisy ECG signal. In the figure, the spike train at the bottom of both (a) and (b) depicts optimal trend switches, as detected by the exact hybrid algorithm. Table II summarizes LS fit and CPU time for $\alpha = 75, 50, 20$.

A couple of important remarks are in order. First, one may verify that streaking is much less pronounced compared with locally monotonic regression, e.g., compare the corresponding regressions for $\alpha = 20$: The first pulse is blotched by locally monotonic regression, yet it is well preserved by piecewise monotonic regression. Second, this reduction in streaking is not at the expense of noise smoothing and/or the ability to follow signal edges in the data, at least for moderate values of α. The drawback is that for very small α, piecewise

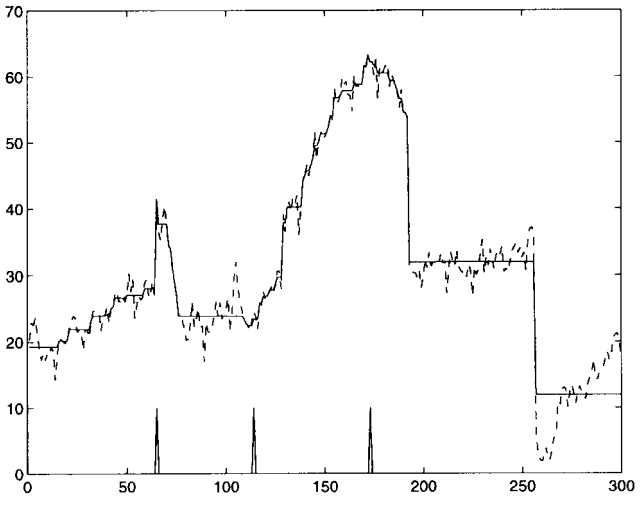


(a)

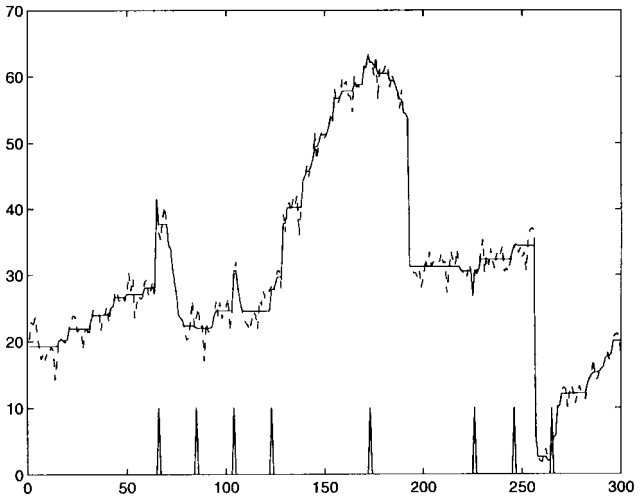


(b)

Fig. 14. (a) Percent improvement in LS fit curves as a function of α. (b) CPU time required by the two-step algorithm as a function of α. Solid curves: $|\mathcal{A}| = 40$. Dashed curves: $|\mathcal{A}| = 20$.



(a)



(b)

Fig. 15. Input (dashed) versus pimo regressions for $\alpha = 50, 20$.

TABLE II
FIT AND CPU TIME (IN MINUTES'SECONDS"-MATLAB ON SUN SPARC10, $N = 300$) FOR PIMO LS REGRESSION

	fit	cputime
$\alpha = 75$	3539	12'20"
$\alpha = 50$	2206	17'11"
$\alpha = 20$	570	35'2"

monotonic regression is more susceptible to outliers than locally monotonic regression.

The choice of ECG signal data was not arbitrary; aside from it exhibiting sharp level transitions (thus being a natural candidate for application of nonlinear smoothing techniques), it was meant to hint on an interesting potential application of piecewise monotonic regression. In the interpretation of ECG signals, the relative timing of the P-wave, QRS-complex, and T-wave is important. These features are relatively easily picked up by a trained eye, even when the data is noisy, yet automatic detection and segmentation is difficult due to changes in heart rate, noise, and other considerations [36]. Consider Fig. 16. It depicts another portion of the same ECG and the result of piecewise monotonic regression of degree $\alpha = 20$. The P-wave, QRS-complex, and T-wave have been manually annotated, and the spike train at the bottom depicts the locations of optimal trend switches, as detected by the exact hybrid algorithm. Notice that P, Q, R, S, and T can be accurately localized by looking at the detected optimal trend switches.

VI. RUNLENGTH CONSTRAINED LEAST SQUARES REGRESSION IN $\mathbb{R}^N$

Let us now consider the runlength constrained least squares problem

$$\begin{aligned} &\textbf{minimize: } \|\mathbf{y} - \mathbf{x}\|_2^2 \\ &\textbf{subject to: } \mathbf{x} \in P_M^N \end{aligned}$$

where P_M^N is the set of all sequences of N elements of $\mathbb{R}$ that are piecewise constant, and the length of constituent pieces is bounded below by M.

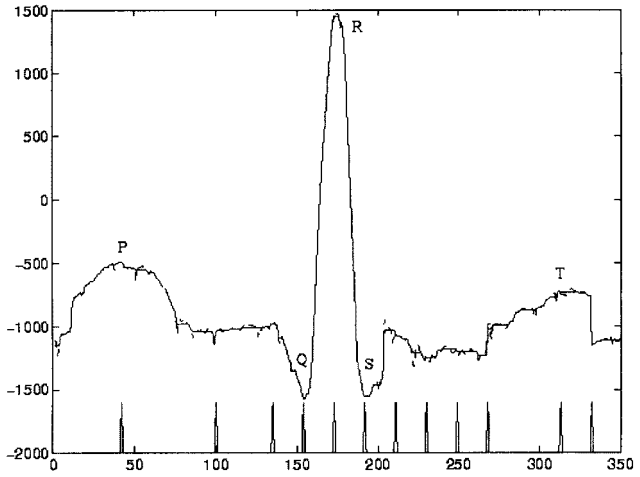


Fig. 16. Piecewise monotonic regression may aid in the detection of significant events in ECG signals. Here, $\alpha = 20$.

This problem appears in the context of segmentation and edge detection. Its digital (finite-alphabet) version has been considered in [5], where an efficient DP algorithm was developed for its solution, and the properties of the associated I/O operator were investigated. Here, we are interested in the same problem but this time in $\mathbb{R}^N$.

We may mimic the development presented for the case of piecewise monotonic regression and associate with each element $\mathbf{x}$ of P_M^N its corresponding *level switching set* $L(\mathbf{x})$. M plays the role of $\alpha - 1$, and the rest of the derivation remains the same. A major simplification is that in runlength constrained regression, the monotone regression subroutine is replaced by simple averaging. We may precompute all required averages and associated costs in-between any two indices at a cost of $O(N^2)$ *before* running the DP program. These precomputed data can be stored in a table for easy access during runtime. Now, replace $\alpha - 1$ with M in the complexity analysis of the proposed algorithm for piecewise monotonic regression. Since all averages and associated costs are stored in the table, $\mathcal{E}^*(t_{i-1}, t_i)$ is readily available for any t_i and t_{i-1}; thus, the computational cost for all required computations for the worst (top) node at stage i is bounded above by $O(N-(i-1)M+1)$. A straightforward calculation shows that the total computational cost for the entire regression is bounded above by $O(\frac{N}{M}[N-M+1] \times [N+1])$. Observe that when $M = N$, this bound predicts complexity $O(N)$, which is exactly what is needed for computing the average of N elements.

A. The LAE Case

We can invoke Corollary 1 to obtain an exact $O(N^3M)$ algorithm for runlength constrained LAE regression in $\mathbb{R}^N$. In addition, the LS algorithm above can be modified to handle runlength-constrained LAE regression in $\mathbb{R}^N$ by replacing the averaging operation in-between level switches by a median calculation. Computational savings may be realized by using an efficient running median algorithm (e.g., the one of Huang *et al.* [37] based on histogram updates) for computing all required constant subregressions.

VII. Oligomodal Regression in $\mathbb{R}^N$

Oligomodal regression is the following problem:

$$\textbf{minimize: } d(\mathbf{y}-\mathbf{x})$$
$$\textbf{subject to: } \pi(\mathbf{x}) = P$$

where $\pi(\mathbf{x})$ is the number of peaks in $\mathbf{x}$, and P is a positive integer (note that a *plateau* is counted as a single peak). This problem is of interest in e.g., chromatography [8], and special problems in density estimation [4].

A. Oligomodal LAE Regression in $\mathbb{R}^N$

Observe that we may determine whether or not $\pi(\mathbf{x}) = P$ by sole knowledge of the sign skeleton of $\mathbf{x}$. In addition, we may construct a DP program to solve a discretized finite-alphabet version of a given oligomodal regression problem. This can be done along the lines of [5] and [6] or, as explained in [8], for an alternative approach to unimodal regression. We may show that the complexity of this program is $O(|\mathcal{A}|^2PN)$, where

- $|\mathcal{A}|$ size of the finite alphabet;
- P number of peaks;
- N total length of the regression.

It therefore follows from Corollary 1 that we may construct a suitable DP program to solve oligomodal LAE regression in $\mathbb{R}^N$ at a complexity cost of $O(N^3P)$.

B. Unimodal LS Regression in $\mathbb{R}^N$

Unimodal regression is the following problem. Given $\mathbf{y} \in \mathbb{R}^N$, find $\mathbf{x} \in \mathbb{R}^N$ to

$$\textbf{minimize: } \|\mathbf{y}-\mathbf{x}\|_2^2$$
$$\textbf{subject to: } \mathbf{x} : \textit{unimodal}$$

i.e., $\pi(\mathbf{x}) = 1$, $\mathbf{x}$ has only one peak. We are particularly interested in non-negative unimodal regression (note that a bounded problem can be transformed into a non-negative problem), in which case, the unimodality constraint can be expressed as ($N = \text{size}(\mathbf{x})$)

$$x(0) \geq 0; \quad x(N) \geq 0; \quad x(n) \geq x(n-1), \quad n = 2, \ldots, j$$
$$x(n) \leq x(n-1), \quad n = j+1, \ldots, N$$

for some mode location j, which *is itself subject to optimization.*

If we fix the mode location, the constrained problem is an instance of quadratic programming (QP). Exhaustive search through all N possible mode locations, each time solving a QP program, gives a first naive way of solving it. Improved algorithms that are more efficient than generic QP for the fixed mode problem, coupled with exhaustive search for the mode location, are discussed in [4]. This problem is of interest in, among other things, chromatographic analysis and flow injection analysis [8]. An interesting result of [8] is a proof that unimodal least squares regression (including optimization of mode location) is no more difficult than two simple Kruskal monotone regressions. This is well under $O(N^2)$ and, in practice, almost $O(N)$: a measurable improvement over exhaustive search for the optimal mode location [4], which is an order of magnitude more complex.

C. Oligomodal LS Regression in $\mathbb{R}^N$

Along the lines of the previously presented hybrid algorithm for piecewise monotonic LS regression, we may envision the construction of a similar algorithm for oligomodal LS regression, consisting of a master DP module and a Kruskal monotone regression (or unimodal regression) module. However, it turns out that even after fixing peak and valley locations, the resulting problem is *not* decomposable in a series of either monotone or unimodal independent subregressions in between these locations, as shown by means of counter example in [8]. Thus, DP will not help solve this LS problem exactly. The point we would like to make is that it may help solve the problem approximately.

An alternative *approximate* approach would be as follows. First, we may construct a DP program to solve a discretized finite-alphabet version of a given oligomodal LS regression problem. This program is a straightforward extension of the unimodal program given in [8]. Note that the resulting DP program provides an exact solution of the discretized problem, which is also an approximate solution of the original problem. As mentioned earlier, the complexity of this program is $O(|\mathcal{A}|^2PN)$. This pseudoregression can be improved by solving a QP problem to compute the optimum regression conditioned on the detected peaks and valleys.

VIII. Conclusion

This paper has focused on the development of efficient algorithms for a class of regression problems that are of interest in nonlinear filtering but in other diverse areas as well. A key result is that under a least absolute error criterion, these problems can be transformed into appropriate finite problems, which can then be efficiently solved via dynamic programming techniques. Although the result does not carry over to least squares regression, hybrid programming algorithms can be developed to solve least squares counterparts of certain problems in the class. As an example, a master/slave DP/monotone regression algorithm has been developed for piecewise monotonic regression.

Future work includes the study of convergence of multiobjective algorithms incorporating some of the present algorithms as subroutines and extensions to handle additional prior knowledge, e.g., smoothness or equality constraints.

Related MATLAB and C programs can be found at http://www.people.virginia.edu/~nds5j and at http://newton.foodsci.kvl.dk/rasmus.html.

Acknowledgment

The work of the first author was motivated in part by earlier work of A. Bovik and A. Restrepo (Palacios). The authors would like to thank the anonymous reviewers for providing constructive feedback, which helped improve this paper.

References

[1] A. Restrepo and A. C. Bovik, "Locally monotonic regression," *IEEE Trans. Signal Processing*, vol. 41, pp. 2796–2810, Sept. 1993.

[2] W. Feller, *An Introduction to Probability Theory and Its Applications (Third Edition—Revised Printing)*. New York: Wiley, 1970, vols. 1 and 2.

[3] A. Restrepo and A. C. Bovik, "Statistical optimality of locally monotonic regression," *IEEE Trans. Signal Processing*, vol. 42, pp. 1548–1550, June. 1994.

[4] R. E. Barlow, D. J. Bartholomew, J. M. Bremner, and H. D. Brunk, *Statistical Inference under Order Restrictions*. New York: Wiley, 1972.

[5] N. D. Sidiropoulos, "The Viterbi optimal runlength-constrained approximation nonlinear filter," *IEEE Trans. Signal Processing*, vol. 44, pp. 586–598, Mar. 1996.

[6] ——, "Fast digital locally monotonic regression," *IEEE Trans. Signal Processing*, vol. 45, pp. 389–395, Feb. 1997.

[7] N. D. Sidiropoulos, J. S. Baras, and C. A. Berenstein, "Weak continuity with structural constraints," *IEEE Trans. Signal Processing*, vol. 45, pp. 3096–3104, Dec. 1997.

[8] R. Bro and N. D. Sidiropoulos, "Least squares regression under unimodality and nonnegativity constraints," *J. Chemometr.*, to be published. Available on-line at http://www.people.virginia.edu/~nds5j.

[9] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.

[10] J. B. Kruskal, "Nonmetric multidimensional scaling: A numerical method," *Psychometrika*, vol. 29, no. 2, pp. 115–129, June 1964.

[11] C. Faloutsos and K.-I. Lin, "FastMap: A fast algorithm for indexing, data-mining, and visualization of traditional and multimedia datasets," *ACM SIGMOD Rec.*, vol. 24, no. 2, pp. 163–174, June 1995.

[12] D. Sankoff and J. B. Kruskal, *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparisons*. Reading, MA: Addison-Wesley, 1983.

[13] A. R. Rao and J. Lohse, "Identifying high level features of texture perception," in *Proc. SPIE 1666*, San Jose, CA, Feb. 1992, pp. 424–435.

[14] B. Marcus, P. Siegel, and J. Wolf, "Finite-state modulation codes for data storage," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 5–37, 1992.

[15] S. Alliney, "A property of the minimum vectors of a regularizing functional defined by means of the absolute norm," *IEEE Trans. Signal Processing*, vol. 45, pp. 913–917, Apr. 1997.

[16] S. G. Tyan, "Median filtering: Deterministic properties," in *Two-Dimensional Digital Signal Processing II: Transforms and Median Filters*, T. S. Huang, Ed.. Berlin, Germany: Springer-Verlag, 1981, pp. 197–217.

[17] N. C. Gallagher, Jr. and G. W. Wise, "A theoretical analysis of the properties of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1136–1141, Dec. 1981.

[18] B. I. Justusson, "Median filtering: Statistical properties," in *Two-Dimensional Digital Signal Processing II: Transforms and Median Filters*, T. S. Huang, Ed. Berlin, Germany: Springer-Verlag, 1981, pp. 161–196.

[19] T. A. Nodes and N. C. Gallagher, "Median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 739–746, 1982.

[20] N. C. Gallagher, Jr., "Median filters: A tutorial," in *Proc. IEEE Int. Symp. Circ., Syst., ISCAS*, 1988, pp. 1737–1744.

[21] A. C. Bovik, T. S. Huang, and D. C. Munson, "A generalization of median filtering using linear combinations of order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 31, pp. 1342–1349, 1983.

[22] A. C. Bovik, T. S. Huang, and D. C. Munson, Jr., "The effect of median filtering on edge estimation and detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 181–194, Mar. 1987.

[23] D. Eberly, H. G. Longbotham, and J. Aragon, "Complete classification of roots of one-dimensional median and rank-order filters," *IEEE Trans. Signal Processing*, vol. 39, pp. 197–200, Jan. 1991.

[24] R. de la Vega and A. Restrepo, "Efficient computation of locally monotonic regression," *IEEE Signal Processing Lett.*, vol. 3, pp. 263–265, Sept. 1996.

[25] J. D. Leeuw, "Correctness of Kruskal's algorithms for monotone regression with ties," *Psychometrika*, vol. 42, pp. 141–144, Mar. 1977.

[26] G. W. Cran, "Amalgamation of the means in the case of simple ordering," *Appl. Statist.*, vol. 29, pp. 209–211, 1980.

[27] G. Bril, R. Dykstra, C. Pillers, and T. Robertson, "Isotonic regression in two independent variables," *Appl. Statist.*, vol. 33, pp. 352–357, 1984.

[28] A. C. Bovik, "Streaking in median filtered images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 493–503, 1987.

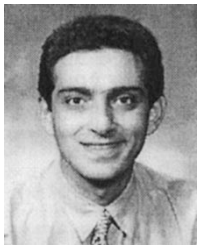
[29] A. Blake, "Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 2–12, Jan. 1989.

[30] R. Bellman, "On the approximation of curves by line segments using dynamic programming," *Commun. ACM*, vol. 4, pp. 284, 1961.

[31] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[32] J. K. Omura, "On the Viterbi decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 177–179, Jan. 1969.

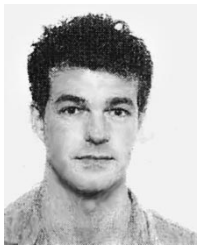
[33] B. Sklar, *Digital Communications*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[34] D. Bertsekas, *Dynamic Programming and Optimal Control Volumes I and II*. Belmont, MA: Athena Scientific, 1995.
[35] S. Haykin, *Communication Systems*, 3rd Ed. New York: Wiley, 1994.
[36] D. H. Brooks and R. S. MacLeod, "Electrical imaging of the heart," *IEEE Signal Processing Mag.*, vol. 14, pp. 24–42, Jan. 1997.
[37] T. S. Huang, G. J. Yang, and G. Y. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 13–18, Jan. 1979.



Nicholas D. Sidiropoulos (M'92) received the Diploma in electrical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park (UMCP), in 1988, 1990, and 1992, respectively.

From 1988 to 1992, he was a Fulbright Fellow and a Research Assistant at the Institute for Systems Research (ISR), UMCP. From September 1992 to June 1994, he served his military service as a Lecturer in the Hellenic Air Force Academy. From October 1993 to June 1994, he was a Member of the Technical Staff, Systems Integration Division, G-Systems Ltd., Athens, Greece. He rejoined ISR-UMCP as a Postdoctoral Research Associate in August 1994 and was promoted to the rank of Assistant Research Scientist in January 1996. He joined the University of Virginia, Charlottesville, as an Assistant Professor of Electrical Engineering in July 1997. His research interests are in the broad area of signal and image processing.

Dr. Sidiropoulos received the NSF/CAREER award (Signal Processing Systems) in June 1998.



Rasmus Bro received the M.Sc. degree in analytical and medical chemistry from the Technical University, Lyngby, Denmark, in 1994.

From 1994 to 1996, he was a Research Assistant (chemometrics/data analysis) at the Chemometrics Group, Food Technology, Department of Food and Dairy Science, Royal Veterinary and Agricultural University, Frederiksberg, Denmark. In the spring of 1995, he spend two months with the Process Analysis Group at the Department of Analytical Chemistry, University of Amsterdam, Amsterdam, The Netherlands. From 1996 to the present, he has worked as a Research Assistant Professor at the Royal Veterinary and Agricultural University in the Chemometrics Group. From October 1997 to March 1998, he worked intermediately as a Guest Researcher at MATFORSK (The Food Research Institute), Aas, Norway. His main research interests are mathematical modeling in a broad sense, mostly associated with so-called multiway modeling and constrained programming.