The MIN PFS problem and piecewise linear model estimation

E. Amaldi§* and M. Mattavelli†

§DEI, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milan, Italy

amaldi@elet.polimi.it

†Integrated System Laboratory (LSI/ISL), Swiss Federal Institute of Technology,

CH-1015 Lausanne, Switzerland. marco.mattavelli@epfl.ch

Abstract

*We consider a new combinatorial optimization problem related to linear systems (*MIN PFS*) that consists, given an infeasible system, in finding a partition into a minimum number of feasible subsystems.* MIN PFS *allows formalization of the fundamental problem of piecewise linear model estimation, which is an attractive alternative when modeling a wide range of nonlinear phenomena. Since* MIN PFS *turns out to be* NP*-hard to approximate within every factor strictly smaller than* $3/2$ *and we are mainly interested in real-time applications, we propose a greedy strategy based on simple randomized and thermal variants of the classical Agmon-Motzkin-Schoenberg relaxation method for solving systems of linear inequalities. Our method provides good approximate solutions in a short amount of time. The potential of our approach and the performance of our algorithm are demonstrated on two challenging problems from image and signal processing. The first one is that of detecting line segments in digital images and the second one that of modeling time series using piecewise linear autoregressive models. In both cases the* MIN PFS*-based approach presents various advantages with respect to classical alternatives, including wider range of applicability, lower computational requirements and no need of* a priori *assumptions regarding the underlying structure of the data.*

[*] Part of this work was performed while E. A. was with the School of Operations Research and Theory Center, Cornell University, Ithaca, NY 14853, USA.

1 Introduction

Although linear models play an important role when modeling a wide range of physical phenomena or technical problems, nonlinear models are often required due to the omnipresence of nonlinearities. Fitting linear models to a set of p data points in n-dimensional space leads to formulations in terms of linear systems such as

$$A\,\mathbf{x} = \mathbf{b} \tag{1}$$

where A is a $p \times n$ real matrix and $\mathbf{b}$ a p-dimensional real vector. The coefficients of each row of A correspond to a data point and each variable x_j, with $1 \leq j \leq n$, to one of the linear model parameters to be estimated. These systems are typically overdetermined, i.e., p is much larger than n. When linear models are too simple to account for the actual complexity of the data, the resulting linear system (1) is infeasible and classical approximate solutions that minimize a least mean square criterion are usually not meaningful. In such situations, piecewise linear models are attractive since they allow approximation of complex nonlinear phenomena but are still simple enough due to local linearity. However, piecewise linear model estimation turns out to be a challenging problem because it involves partitioning the data points into disjoint subsets as well as fitting a linear submodel to each one of these subsets. The structure and parameter estimation appears as a *chicken and egg problem* in the sense that if one of them is known the other one can be determined relatively easily.

Overdetermined infeasible systems are often tackled with robust regression techniques [35], which assume that most of the data points can be fitted by a linear model and that the system (1) is infeasible due to spurious or noisy data. But these methods have a limited range of applicability since they can cope with at most 50% of outliers, that is, a linear submodel must fit at least half of the data points.

Two-phase approaches in which one first partitions the data using some clustering methods [21] and then estimates the parameters of each submodel using robust regression techniques [35] have major drawbacks. In particular, the number of linear submodels needs to be guessed in advance and the clustering problems, which are NP-hard, are difficult to solve. Even more importantly, the partitions obtained during the clustering phase may lead to meaningless results. Indeed, standard clustering methods, which minimize (maximize) some dissimilarity (similarity) measure, do not take into account the type of submodels (linear) used to fit the data locally. For a survey of mathematical programming approaches to clustering problems see [16] and the references herein.

In principle, the Hough Transform (HT) and its variants [18, 22] constitute a good alternative to the two-phase approach because they can solve a wide range of linear and nonlinear model identification problems without requiring any a priori assumption. Local information is used to accumulate evidence for some particular sets of parameter values of the model under consideration. Being relatively insensitive to noise and partially incorrect data, HT variants have been applied to a number of image and signal processing problems (see [22] and refereces herein). However, they have high computational requirements to guarantee a reasonable accuracy and they are quite sensitive with respect to threshold settings.

In this paper we present a new combinatorial optimization approach to simultaneously determine the structure and estimate the parameters of piecewise linear models. In Section 2 we show that this problem can be formulated as that of Partitioning linear systems into a MINimum number of Feasible Subsystems. This combinatorial optimization problem, which we refer to as MIN PFS, is shown in Section 3 to be NP-hard to approximate within every constant factor $\rho < 3/2$. In Section 4 we describe a simple greedy strategy which provides good approximate solutions in a short amount of time. The algorithm is based on randomized and thermal variants [2, 3, 12] of the classical Agmon-Motzkin-Schoenberg relaxation method [1, 30] for solving systems of linear inequalities. In Section 5 the performance of the MIN PFS-based method are demonstrated by reporting some results for two challenging problems from image and signal processing. In particular, we focus here on the fundamental problem of detecting line segments in digital images and that of time series analysis using piecewise linear autoregressive models. Differences with respect to conventional alternatives such as robust regression, clustering methods and the Hough transform are mentioned. Finally, Section 6 contains some concluding remarks.

2 From the MIN PFS problem to piecewise linear model estimation

To allow formalization of the fundamental problem of piecewise linear model estimation, we consider the following combinatorial optimization problem which, to the best of our knowledge, has not yet been introduced in the discrete mathematics literature.

> MIN PFS: Given a possibly infeasible linear system $A\mathbf{x} = \mathbf{b}$ with $A \in \mathbf{R}^{p\times n}$ and $\mathbf{b} \in \mathbf{R}^{p}$, find a Partition of this system into a MINimum number of Feasible Subsystems.

The MIN PFS problem, which is of interest in itself, is very attractive in this context because it provides a natural way of addressing simultaneously the two fundamental issues: data partition and parameter estimation. Given any solution of MIN PFS the partition of the equations indicates the data partition and a solution of each feasible subsystem provides a set of parameter values for the corresponding submodel. Unlike in two-phase approaches, linearity of the submodel is here taken into account in the partition process.

A unique feature of our MIN PFS formulation is that it aims at minimizing the number of submodels. According to the well-known Occam razor principle [8], we look for the "simplest" piecewise linear model consistent with the data, which is most likely to be the correct one. Here complexity is measured in terms of the number of linear submodels. The choice of the objective function clearly tends to penalize irrelevant linear submodels that account for just a few (possibly spurious) data points. Note that here the partition is not determined based on the distances between the data points in n-dimensional space. Points are grouped together only if they can be fitted by the same linear submodel or equivalently if the corresponding equations in (1) form a feasible subsystem. It has also to be emphasized that no a priori assumptions are made concerning the type of piecewise linear model. In particular, the corresponding linear submodels are not restricted to define a continuous or single-valued function.

In practice, to cope with noisy data, a maximum noise tolerance threshold $\varepsilon > 0$ is selected and each equation $\mathbf{a}^{k}\mathbf{x} = b_k$, where $\mathbf{a}^k$ denotes the kth row of A and b_k the kth component of $\mathbf{b}$ with $1 \leq k \leq p$, is replaced with the two complementary inequalities:

$$\mathbf{a}^k\mathbf{x} \leq b_k + \varepsilon \qquad \mathbf{a}^k\mathbf{x} \geq b_k - \varepsilon \tag{2}$$

that have to be simultaneously satisfied. This standard appproach is widely used in the context of feasible linear systems, see for instance [10]. If equations of the original system are expected to be affected by different noise levels, different thresholds can of course be used.

Note that this variant of MIN PFS has a simple geometric interpretation in coefficient space. As shown in Figure 1, if $b_k = 0$ for all $1 \leq k \leq p$ it amounts to finding the minimum number of hyperslabs of thickness 2ε (obtained by moving a hyperplane $\{\mathbf{a} \in \mathbf{R}^n | \mathbf{a}\mathbf{x} = 0\}$ defined by $\mathbf{x} \in R^n$ with $||\mathbf{x}|| = 1$ apart by ε in both opposite directions) such that all the points $\mathbf{a}^k \in \mathbf{R}^n$ representing the coefficient vectors are contained in at least one hyperslab. The normal vectors defining these hyperplanes clearly correspond to the solutions of the feasible subsystems in a minimum partition. In the general case with arbitrary right-hand sides b_k, the hyperplanes are not required to go through the origin and the geometric interpretation is slightly more involved.

Finally, it is worth pointing out the connection between MIN PFS and the classical problem of minimum node colouring in graphs.

> MIN GRAPH COLOURING: Given a graph $G = (V, E)$, colour the nodes with as few colours as possible so that adjacent nodes have different colours.

Property: MIN PFS with equations and $\{0, 1\}$-variables admits MIN GRAPH COLOURING as a particular case.
Given any instance of the latter problem defined by $G = (V, E)$ we construct the following special instance of MIN PFS. For each $v_i \in V$ of $G = (V, E)$, we consider the equation

$$x_i - \sum_{j:[v_i, v_j] \in E} x_j = 1.$$

If $E \neq \emptyset$, the resulting system is infeasible. Since feasible subsystems are in one to one correspondence with independent sets, partitions into s feasible subsytems are in one to one correspondence with colourings using s different colours.

Thus MIN PFS with real variables is of particular interest from the piecewise linear model point of view, while the version with binary variables generalizes a well-known problem on graphs.

3 Worst-case complexity of MIN PFS

Suppose, without loss of generality, that there are no pairs of contradictory equations, e.g., equations with the same coefficient vectors $\mathbf{a}^k$ and a different b_k. If the $\mathbf{a}^k$ are in general position[1], the problem is trivial since any subsystem with up to n equations is feasible and all larger subsystems are infeasible. However, the problem turns out to be harder in general. We say that MIN PFS can be approximated within a factor $\rho \geq 1$ if there exists an algorithm that is guaranteed to deliver for any instance a partition into feasible subsystems with at most ρ times more subsystems than in a minimum size partition.

Theorem 1 MIN PFS *is* NP-*hard and cannot be approximated within any constant factor* $\rho < 3/2$*, unless* P $=$ NP.

PROOF We show that it is NP-complete to decide whether infeasible systems can be partitioned into two feasible subsystems. The proof is by reduction from the following classical NP-complete problem, see [13].

> PARTITION: Given a finite set $S = \{1, \dots, n\}$ and a size s_j for each $j \in S$, does there exist a subset $S' \subseteq S$ such that $\sum_{j \in S'} s_j = \sum_{j \in S - S'} s_j$?

Since linear system feasibility can be tested in polynomial time, the decision version of MIN PFS in which one ask whether any given instance $(A, \mathbf{b})$ is partitionable into two feasible systems is clearly in NP.

Let $(S, \{s_1, \dots, s_n\})$ be an arbitrary instance of the PARTITION problem. We construct a particular instance $(A, \mathbf{b})$ of MIN PFS such that the answer to the former one is affirmative (i.e. there exists a subset such that $\sum_{j \in S'} s_j = \sum_{j \in S - S'} s_j$) if and only if the answer to the latter one is also affirmative (i.e. it is possible to find a partition into two feasible subsystems).

The idea is to construct an infeasible system containing one variable x_j for each $j \in S$. The system is composed of the equation

$$\sum_{j=1}^{n} s_j x_j = 0 \tag{3}$$

as well as of the $2n$ equations

$$x_j = 1 \quad \text{for} \quad j = 1, \dots, n, \tag{4}$$

$$x_j = -1 \quad \text{for} \quad j = 1, \dots, n. \tag{5}$$

Since for each j, $1 \leq j \leq n$, the equations of types (4) and (5) are contradictory, the overall system with $2n+1$ is infeasible. Clearly, such a system can always be partitioned in at most three feasible subsystems: one for equation (3), one for the equations of type (4) and one for those of type (5).

[1] A set of points in $\mathbf{R}^n$ is in general position if no subset of $k+1$ points is contained into an affine space of dimension $k-1$.

If the considered instance of PARTITION has a positive answer and $(S', S - S')$ is any bipartition of S such that $\sum_{j \in S'} s_j = \sum_{j \in S - S'} s_j$, the vector $\mathbf{x}$ defined by

$$x_j = \begin{cases} 1 & \text{if } j \in S' \\ -1 & \text{otherwise} \end{cases}$$

satisfies equation (3) and exactly n equations of types (4) and (5). The other n equations of types (4) and (5) are obviously feasible. The overall system can thus be partitioned into two feasible subsystems, one containing $n + 1$ equations and the other one n equations.

Conversely, suppose that the system (3)–(5) can be partitioned into two feasible subsystems and that $\mathbf{x}^1$ and $\mathbf{x}^2$ are two corresponding solutions. By construction, each of them satisfies exactly n equations of types (4)–(5) and hence has -1 or 1 components. If $\mathbf{x}^1$ denotes the solution which satisfies also equation (3), the bipartition of S defined by $j \in S'$ if and only if $x^1_j = 1$ satisfies the relation $\sum_{j \in S'} s_j = \sum_{j \in S - S'} s_j$.

It is easy to see that this reduction also implies that it is NP-hard to find near-optimal solutions within the above-mentioned factor. Suppose there exists a polynomial time algorithm that is guaranteed to yield for every instance a partition with a number of feasible subsystems which exceeds the minimum one by a multiplicative factor $\rho < 3/2$. Then for any bipartitionable instance it would yield a solution of size strictly less than $3/2 \cdot 2 = 3$, that is, with 2 feasible subsystems. Hence the contradiction since such an algorithm would solve an NP-complete problem in polynomial time.

Although this worst-case result provides insight into the inherent difficulty of MIN PFS, it does not rule out the existence of efficient heuristics with a good average-case behavior. Now, in a number of applications, including the two discussed in Section 5, finding reasonably good approximate solutions in a short amount of time is of great practical value.

Note that MIN PFS is trivial for infeasible systems of inequalities since any such system can be partitioned into two feasible subsystems. This fact is obvious for homogeneous systems and is easily verified for inhomogeneous ones [36]. But the above reduction can be extended to the case of systems of complementary inequalities like (2) in which such pairs must belong to the same feasible subsystem.

Thus estimating piecewise linear models turns out to be more difficult than estimating linear models which can be done in polynomial time using least mean square methods.

4 A greedy algorithm

Since in various practical applications we are interested in finding approximate solutions of MIN PFS rapidly, we adopt a simple greedy approach and subdivide the overall partition problem into a sequence of subproblems. Starting with the infeasible system composed of pairs of complementary Inequalities (2) corresponding to a single linear model that is not powerful enough to model the data at hand, we iteratively extract close-to-maximum feasible subsystems, i.e., feasible subsystems containing a close-to-maximum number of pairs of complementary inequalities. Clearly, iterating the process until the remaining subsystem is feasible yields a partition into feasible subsystems. As we shall see in Section 4.2, this greedy strategy is well motivated from the application point of view and it turns out to be experimentally effective, among others, in the applications discussed in Section 5.

4.1 Extraction of close-to-maximum feasible subsystems

At each step of the greedy strategy, the subproblem to deal with is that of, given the current system $A\mathbf{x} = \mathbf{b}$ and a tolerance threshold $\varepsilon > 0$, seeking an $\mathbf{x} \in \mathbf{R}^n$ that satisfies as many pairs of complementary inequalities $\mathbf{a}^k\mathbf{x} \leq b_k + \varepsilon$ and $\mathbf{a}^k\mathbf{x} \geq b_k - \varepsilon$ as possible. This problem is an extension to the setting with pairs of inequalities of the combinatorial optimization problem of finding a Maximum Feasible Subsystem of infeasible linear inequality system, which is known in the literature as MAX FS [6]. Weighted and unweighted versions of MAX FS have a number of interesting applications in various fields such as computational geometry, operations research, radiation therapy and machine learning (see e.g. [2, 11, 23, 24, 33] and the included references). As shown in [4] MAX FS can be approximated within a factor of 2 but it does not admit a polynomial-time approximation scheme, unless P = NP.

Given the suboptimality of the greedy strategy, it is neither necessary nor desirable to look at each step for an optimal solution of the corresponding subproblem, see the discussion in Subsection 4.2. We are interested in algorithms that provide good solutions in a reasonably short amount of computation time. In this paper we propose to tackle the above extension of MAX FS using randomized and thermal variants of the classical Agmon-Motzkin-Schoenberg (AMS) relaxation method for solving systems of linear inequalities [1, 30]. Variants of the AMS procedure have been investigated since the 60's in the machine learning literature under the name of percetron-like methods, see e.g. [29]. The randomized and thermal relaxation method used here is a variant of those studied in [2, 3], which extend the thermal perceptron procedure [12] [2]. In spite of the inherent computational complexity of MAX FS these methods perform very well in practice. Deterministic AMS-like relaxation methods have also been extensively studied in the mathematical programming literature, often as special cases of subgradient algorithms, see e.g. [10, 14]. The adavantages of randomization for a similar relaxation method for feasible systems of convex inequalities have also been investigated in [34].

Given a feasible system $A\mathbf{x} \leq \mathbf{b}$, the AMS method is a simple iterative procedure that generates a sequence of estimates. Starting with an arbitrary initial guess $\mathbf{x}^0 \in \mathbf{R}^n$, at each iteration an inequality is selected according to a prescribed rule (e.g., cyclical choice, inequality with maximum violation or uniformly at random among the p ones) while all the others are relaxed. Suppose that at iteration i the kth inequality $\mathbf{a}^k\mathbf{x} \leq b_k$ is considered. Then the current estimate $\mathbf{x}^i$ is updated as

[2] The adjective "thermal" refers to the fact that a temperature parameter is used like in simulated annealing.

follows:

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \delta_i \mathbf{a}^k \tag{6}$$

where $\delta_i = \max\{0, \lambda_i(\mathbf{a}^k\mathbf{x}^i - b_k)\}$ and $\lambda_i > 0$. In other words, the current estimate is updated only if it violates the inequality at hand.

Geometrically, the current estimate $\mathbf{x}^i$ can be viewed in parameter space as a point in $\mathbf{R}^n$ and the kth inequality defines a hyperplane $H_k = \{\mathbf{x} \in \mathbf{R}^n | \mathbf{a}^k\mathbf{x} = b_k\}$. If $\mathbf{x}^i$ lies on the negative side of H_k (kth inequality is satisfied) then it is unchanged. Otherwise, $\mathbf{x}^{i+1}$ is obtained by making a step along the line defined by the extremity of $\mathbf{x}^i$ and that of its orthogonal projection onto H_k. The steplength clearly depends on the value of λ_i, see Figure 2.

For any feasible system, if the increment sequence $(\delta_i)_{\mathbf{N}}$ satisfies certain conditions then the AMS method is guaranteed to yield a solution that satisfies all inequalities in a finite number of iterations but in the worst-case exponential number of iterations, see e.g. [14, 37]. Although more sophisticated stopping criteria have been proposed [37], in practice one usually stops after a predefined maximum number of cycles C through the p inequalities. For infeasible systems, the procedure never terminates (violated inequalities always trigger updates) and in a worst-case scenario it can even lead in a single iteration from an estimate that satisfies a maximum number of inequalities to a worst possible one.

Based on [2, 12], we propose to look for maximum feasible subsystems of $A\mathbf{x} \leq \mathbf{b}$ using a Thermal variant of the above Randomized AMS Relaxation method, referred to as TRR method. To make the procedure well behaved for infeasible systems, the basic idea is to favor updates of the current estimate $\mathbf{x}^i$ which aim at correcting unsatisfied inequalities with a relatively small *violation* $v_i^k = \mathbf{a}^k\mathbf{x}^i - b_k$ if $\mathbf{a}^k\mathbf{x}^i > b_k$ and 0 otherwise. Indeed, the large modifications that would be required to correct unsatisfied inequalities with large v_i^k are likely to corrupt other inequalities that $\mathbf{x}^i$ satisfies. As in [12], it is appropriate to pay decreasing attention to unsatisfied inequalities with large violations, namely, to perform at the beginning all updates prescribed by the standard AMS procedure and then only those which aim at correcting unsatisfied inequalities with progressively smaller and smaller v_i^k. This can be achieved by letting the magnitude of the updates δ_i decrease exponentially with v_i^k and by introducing a control parameter T with which the v_i^k are compared. More specifically, $\delta_i = \frac{T_i}{T_0} \cdot \exp\left(-\frac{|v_i^k|}{T_i}\right)$ where T_i is the *temperature* value at the ith iteration and the sequence $(T_i)_{\mathbf{N}}$ is reduced (asymptotically or over a finite number of iterations) from an initial value T_0 to 0. The approximate solution provided by the algorithm is the best estimate $\mathbf{x}^i$ generated during the process, i.e., the one that satisfies the largest number of inequalities.

As pointed out in [3], TRR versions are fully specified by: the *selection rule* which indicates the way the inequalities are selected (e.g. uniformly at random *with* or *without* replacement), the *increment rule* specifying how the δ_i are computed (e.g. $\delta_i = 1$ or linearly/exponentially reduced over a prescribed number of iterations) and the *decision rule* characterized by the probability π_i with which the update (6) is performed at the ith iteration. While in the basic version generalizing the thermal perceptron $\pi_i = 1$ for all unsatisfied inequalities, in the doubly randomized thermal variants [3] an update is performed with a given probability π_i that depends on the degree of violation of the inequality at hand. For instance, we may take $\pi_i = \frac{T_i}{T_0} \cdot \exp\left(-\frac{|v_k^i|}{T_i}\right)$ or $\pi_i = \exp\left(-\frac{|v_i^k|}{T_i}\right)$ and, respectively, $\delta_i \equiv 1$ or $\delta_i = T_i/T_0$. Instead of updating $\mathbf{x}^i$ by small rapidly decreasing amounts with probability $\pi_i = 1$ for all unsatisfied inequalities, the idea is to use a constant or more slowly decreasing increment δ_i, but to perform the updates with rapidly decreasing probability.

Although the thermal perceptron procedure is contrasted in [12] with gradient-type methods, there is a closely related interpretation of the TRR method [3]. For any fixed value of the temperature T, consider the function $\chi_T : \mathbb{R} \to \mathbb{R}$ that takes the value $1 - \exp(-v/T)$ for nonegative $v \in \mathbb{R}$ and is zero on the negative half-line. χ_T approximates the indicator function of the positive half-line. The function

$$J_T(\mathbf{x}) = \sum_{\{k:\mathbf{a}^k\mathbf{x}>b_k, 1\le k\le p\}} \chi_T(\mathbf{a}^k\mathbf{x} - b_k) \tag{7}$$

is thus a piecewise smooth approximation to the objective function J_0 that counts the number of misclassified inequalities, and J_T converges pointwise to J_0 when $t \to 0$. Note that

$$\nabla_s J_T(\mathbf{x}) := \sum_{\{k:\mathbf{a}^k\mathbf{x}>b_k, 1\le k\le p\}} \frac{1}{T}\exp\left(-(\mathbf{a}^k\mathbf{x} - b_k)/T\right)\mathbf{a}^k \tag{8}$$

is a subgradient of J_T, and that the expected update of the TRR variant that selects each inequality with equal probability $\pi = 1/p$ at each iteration is equal to the dampened subgradient step

$$E[\Delta\mathbf{x}] = -\frac{T^2}{pT_0}\nabla_s J_T(\mathbf{x}) \tag{9}$$

with the damping factor $\frac{p \cdot T^2}{T_0}$. The TRR method may thus be interpreted as a *randomized subgradient homotopy method*. For large values of T, J_T is a coarse approximation of the actual objective function J_0 we would like to minimize. When T approaches 0, the approximation quality progressively increases and strong damping is needed in order to avoid instabilities. This suggests that the value of T_0 as well as the way the T is reduced may affect the quality of the final solutions as it is, for instance, the case in simulated annealing.

In [3] we establish almost sure finite convergence guarantees to an optimal solution of MAX FS for several variants of the TRR procedures. For instance, for the doubly randomized TRR method with $\pi_i = T_i/T_0 \exp\left(-|v^k|/T_i\right)$ and $(\delta_i)_{\mathbb{N}} \equiv 1$, we prove that when applied to any given strictly feasible linear inequality system then for a large enough constant $c > 0$ (which depends on $\mathbf{x}^0$ and on the system coefficients) and a temperature schedule that asymptotically decreases no faster than $T_i = c/\ln i$, for $i \geq 2$, then there exists some index $i_0 \in \mathbb{N}$ such that with probability 1 the best solution found up to iteration i_0 is optimal.

From a practical point of view, it is worth emphasizing, as it is done in [10] for feasibile linear systems, that our AMS variants are inherently parallel and hence well-suited to handle even extremely large (infeasible) systems.

The TRR method is easily adapted to the case with pairs of complementary Inequalities (2). Starting with an arbitrary initial $\mathbf{x}^0 \in \mathbf{R}^n$, at each iteration a pair of inequalities is selected according to a prescribed rule (e.g., uniformly at random without replacement). If the current estimate $\mathbf{x}^i$ does not satify both them, the *violation* is defined as follows:

$$v_i^k = \begin{cases} \mathbf{a}^k\mathbf{x}^i - b_k - \varepsilon & \text{if } \mathbf{a}^k\mathbf{x}^i \geq b_k + \varepsilon \\ b_k - \varepsilon - \mathbf{a}^k\mathbf{x}^i & \text{if } \mathbf{a}^k\mathbf{x}^i \leq b_k - \varepsilon \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The proposed algorithm is described below. T is the temperature parameter which is linearly (or exponentially) decreased from an initial value T_0 to 0 in the predefined maximum number of cycles C. For instance, at the beginning of the c^{th} cycle $T = \gamma T_0$ with $\gamma = 1 - \frac{c}{C}$. See the Appendix for more details on this annealing schedule, including an appropriate choice of T_0.

Algorithm 1

```
Initialization: Pick any $\mathbf{x}^0 \in \mathbf{R}^n$, set cycle counter $c = 1$, select maximum number of cycles
$C$ and initial $T_0$
while $c \leq C$ do
    Initialize set of indices $I = \{1, \ldots, p\}$
    repeat —cycle through all pairs of inequalities—
        Pick index $k_i$ equiprobably and without replacement from $I$
        Compute violation $v_i^{k_i}$ for the $k_i$th pair of inequalities
        Set $T = (1 - \frac{c}{C}) \cdot T_0$ and $\delta_i = \frac{T}{T_0} \exp\left(\frac{-v_i^{k_i}}{T}\right)$
        if $\mathbf{a}^{k_i}\mathbf{x}^i \geq b_{k_i} + \varepsilon$ then $\mathbf{x}^{i+1} = \mathbf{x}^i - \delta_i\,\mathbf{a}^{k_i}$
        else if $\mathbf{a}^{k_i}\mathbf{x}^i \leq b_{k_i} - \varepsilon$ then $\mathbf{x}^{i+1} = \mathbf{x}^i + \delta_i\,\mathbf{a}^{k_i}$
            else $\mathbf{x}^{i+1} = \mathbf{x}^i$
        Set $I = I - \{k_i\}$
    until $I = \emptyset$
    Update $T_0$ and set $c = c + 1$
end
return $\mathbf{x}^{p \cdot C}$
```

Ideally one would like to return as an approximate solution the best estimate generated during the C cycles. However, it is very time consuming to check for each new estimate $\mathbf{x}^{i+1}$ whether it satisfies a larger number of pairs of inequalities (2) than the best one generated so far. Therefore this explicit test is only performed when the number of consecutive randomly picked pairs of inequalities that have been satisfied by the current estimate is larger than the number of those correctly satisfied by the best solution encountered so far before it was updated.

Note that, unlike in regression methods, inequalities with a large violation with respect to the current temperature T yield small modifications of the current estimate and those that are satisfied do not yield any modification. For very high values of T, any violated inequality yields a significant update, while for low values, only those with small violations with respect to T yield significant updates.

4.2 Properties of the greedy strategy

In principle, implicit enumeration methods, like the general technique for maximizing functions of linear relations presented in [15], could be adapted to find optimal solutions in a finite but in the worst-case exponential amount of time. However, given the suboptimality of the greedy strategy, it is neither necessary nor desirable to look at each step for a largest feasible subsystem. The following simple example shows that even if maximum feasible subsystems are available at each step, the greedy strategy is not guaranteed to yield minimum partitions. The cardinality of the resulting partitions may even depend on the order in which the maximum feasible subsystems are extracted since some equations may be assigned to different feasible subsystems.

Example 1 Consider the system:

$$\begin{aligned} x_1 + x_2 &= 0 && (11)\\ x_1 - x_2 &= 0 && (12)\\ x_2 &= 1 && (13)\\ x_2 &= 2 && (14) \end{aligned}$$

whose largest feasible subsystems include two equations. Clearly, there are five corresponding solutions: $(0,0)$ satisfies equations (11-12), $(1,1)$ equations (12-13), $(2,2)$ equations (12) and (14), $(-1,1)$ equations (11) and (13) and $(-2,2)$ equations (11) and (14). If one starts with the maximum feasible subsystem composed of equations (11-12), then two additional subsystems are needed in the partition because the two remaining equations are infeasibble. Now a minimum partition includes only two feasible subsystems, for instance equations (12) and (14) on one side and equations (11) and (13) on the other side.

In practice this kind of ambiguity, which is inherent to the data (some data points may be consistent with several submodels), can be resolved through an appropriate postprocessing stage based on additional application specific information. In this work, each data point is assigned to the submodel which gives rise to the minimum equation violation.

The idea of breaking down the original problem into that of determining a sequence of close-to-maximum feasible subsystems is particularly attractive in applications such as image and signal processing where the ultimate goal is to achieve one-line processing at the lowest possible implementation costs.

From an application point of view it really makes sense to look first for the dominant submodels, i.e., those which account for the largest fraction of the data points. This corresponds to the sequential paradigm mentioned in [16] for clustering problems. Clearly the size of the feasible subsystems corresponding to actual submodels is typically much larger than the number of variables n. In the presence of noisy or spurious data, once the actual structure has been extracted the points that are left are usually in general position and they can be partitioned into feasible subsystems of size at most n. Such feasible subsystems can be clearly discarded since they do not contain any significant modeling information. Thus, the exact minimum partition is not required and a greedy heuristic which looks first for the largest feasible subsystems is appropriate.

An important feature of our approach is that it provides an estimate of the number of linear submodels needed to fit the data. Once this information is available other formulations and techniques can be used to further improve the quality of fit. For instance, to achieve higher robustness for noisy data, a least mean square solution can be determined for each subsystem of the partition provided by our algorithm. As an alternative one may also try to minimize the sum of squares of the 2-norm distances between each data point and the nearest hyperplane among those associated to the given number of submodels, see the heuristic in [9] for minimizing a concave function formulation on a polyhedral set.

5 Two applications in image and signal processing

To demonstrate the wide applicability of our MIN PFS-based approach and the performance of our greedy algorithm, we describe how they can be adapted to tackle two challenging problems from image and signal processing. The first one is the fundamental problem of detecting line segments in digital images and the second one that of modeling time series using piecewise linear autoregressive models. Here we focus on the distinctive features of our method and report some typical results, more details can be found in the application-oriented papers [26, 27, 28, 41].

5.1 Line detection in digital images

Suppose we are given a $m \times l$ digital image with a grey level for each pixel and suitable procedures are used to extract the contour points. A central problem in image processing that has been extensively studied in the literature is that of detecting line segments in contour point images, where pixels take value 1 or 0 depending on whether they correspond to contour points (see for instance [43] and references herein).

The problem of classifying a set of contour points $\{(a_1^k, a_2^k)\}_{1 \leq k \leq p}$ of an image into line segments can be naturally formulated in terms of MIN PFS. Since the coordinates of all contour points (a_1^k, a_2^k) lying on a same straight line satisfy a linear equation

$$a_1^k x_1 + a_2^k x_2 + x_3 = 0, \tag{15}$$

where x_j, $1 \leq j \leq 3$, are the line parameters and only two of them are actually independent, it suffices to construct a linear system $A\mathbf{x} = \mathbf{1}$ with a row for each contour point and two variables corresponding to the two parameters needed to define a line in 2-D. More specifically, a_{1k} and a_{2k} correspond to the first and, respectively, the second coordinates of the kth contour point. Note that deleting the third parameter x_3 and including the right hand side $\mathbf{1}$ amounts to exclude only the lines that exactly go through the origin. In the presence of several line segments and noise in the image or from the contour point extraction process, the resulting linear system is infeasible. Conversely, any feasible subsystem corresponds to a subset of contour points that lie on the same straight line, and any partition into s feasible subsystems amounts to a partition of all contour points into s line segments. The solution of each subsystem provides the parameters of the corresponding line. Given the objective function of MIN PFS, we look for the smallest set of line segments that account for all contour points. To cope with noise and quantization errors, it suffices to replace each equation by the two complementary Inequalities (2). An appropriate value of the parameter $\varepsilon > 0$ can be determined by applying the experimental procedure described in Subsection 5.2.1. The resulting value clearly depends on the image resolution and the desired line detection accuracy.

Although this simple application of the MIN PFS-based approach to line detection is quite elegant and powerful, the TRR procedure tends to converge slowly for special lines with close to infinite slope $-x_1/x_2$ or lying within a small distance from the origin. Significant speedup can, however, be obtained by applying the same type of procedure to an extended variable space. In [27], for instance, a three dimensional space over the variables x_j, $1 \leq j \leq 3$, of the equations $a_1^k x_1 + a_2^k x_2 + x_3 = 0$ is considered. To avoid convergence to the all zeroes trivial solution of the homogenous system $\mathbf{Ax} = \mathbf{0}$, the current solution $\mathbf{x}^i$ is projected, at each relaxation iteration, on a 3-D surface (e.g., a cylinder or a sphere) that does not contain the trivial solution (the origin) but that contains at least one of the infinite solutions x_j, $1 \leq j \leq 3$, corresponding to the same solution in the 2-D parameter space. More details on this convergence issue and application can be found in [26, 27, 28]. Notice that regardless of whether it is applied to the *natural* or *extended* solution space the greedy MIN PFS-based approach is the same except for the above-mentioned projections in the TRR procedure.

5.1.1 Differences with respect to the Hough Transform approach

Before presenting some typical results, we point out the main differences between our approach and the variants of the Hough Transform (HT) that are extensively used in the literature [20, 43].

In the classical HT, each contour point (feature in general) is mapped onto a line (hyperplane) in the parameter space corresponding to all combinations of the parameter values consistent with it, i.e., all straight line segments passing through that contour point. Attention is restricted to a limited region of the parameter space which is subdivided into 2-D cells (hypercells). The goal is then to identify a cell (hypercell) in this accumulation array that is hit by the largest number of lines (hyperplanes), i.e., sets of parameter values that account for the largest number of contour points. Such peaks are detected by using exhaustive (see Figure 3) or reduced search strategies. The parameter estimation accuracy clearly depends on the quantization, namely, the size of the cells (hypercells). In principle, the HT enables estimation of piecewise linear models without any *a priori* assumption. However, the fine resolution needed to guarantee a reasonable accuracy usually requires very large amounts of memory and computation time.

A number of variants, including the hierarchical, probabilistic, robust and randomized HT [31], have been developed to try to overcome these major drawbacks. But unfortunately these refined search strategies (see [18, 22]) which considerably reduce in some cases the computational requirements are not appropriate when several models (line segments) have to be simultaneously identified (e.g., the submodels of a piecewise linear model) or when there is a relatively high level of noise. [20, 43]. In the Randomized HT [42] (RHT), for instance, one may have to sample a much larger number of subsets of contour points in order to achieve the ad-hoc threshold indicating the presence of a probable peak. In coarse to fine strategies such as the adaptive HT [17], the initial stages where a coarse quantization is considered may fail to detect some of the peaks corresponding to the submodels to be identified. Indeed, relatively small peaks may be hidden in a noisy background. In general, selecting small threshold values may yield erroneous solutions, while larger values may substantially increase the computational load and, therefore, jeopardize the reduced time complexity and lower memory requirements. Thus, when several line segments have to be detected, there is a very delicate trade-off between time/memory requirements and solution quality.

It is interesting to note that our TRR procedures can be viewed as an algorithm to search for peaks in a continuous accumulation space. Although continuous kernels can be used in HT variants [32], they just allow to refine the accuracy of the parameter estimation after the standard accumulation and search stages.

Our MIN PFS-based method and the HT variants differ considerably in terms of computational complexity. The former has negligible memory requirements (it just stores contour points and their assignment to subsystems) and its computational complexity is entirely characterized by arithmetic processing. On the contrary, HT variants are based on simpler processing steps (counting operations) but they have very heavy memory requirements, since they require storing the accumulation array that needs to be scanned (hypercell by hypercell) for each data point. For isotropic quantizations, the number of hypercells is given by r^n, where r is the number of intervals along each parameter space dimension and n is the number of parameters [25].

In the 2-D line detection case, if the image and accumulation array resolutions are increased accordingly, both memory requirements and time complexity increase with the square of the resolution increase factor. In contrast, the time complexity of the MIN PFS-based method only depends on the number of contour points and the number of feasible subsystems to be extracted. Notice that the number of contour points on a line increases linearly with the image resolution. Since line segments are detected in an sequential way, starting from the dominant one, our algorithm can be clearly stopped as soon as the relevant features are detected. In applications such as object tracking, this can lead to an additional substantial time complexity reduction while the HT variants need to compute the entire Hough transform before providing the desired results.

It is worth emphasizing that the MIN PFS-based approach can also be generalized to detect higher order curves or surfaces, e.g., circles and ellipses. In such cases the advantages versus the HT become dramatic. For circles and ellipses, for instance, HT complexity increases with the power of s, where s is the number of free parameters defining a circle ($s = 3$) or an ellipse ($s = 5$).

To conclude, our MIN PFS-based approach is particularly suited for efficient implementation on simple floating point Digital Signal Processors (DSPs). Moreover, given the general trend in the progress of hardware performance (much faster increase in processing capabilities than in memory access speed or memory capacity), its considerable computational advantages with respect to HT variants should become even more substantial in the future.

5.1.2 Some typical results

We report here some comparative results obtained with our MIN PFS method and with the standard as well as randomized HT (RHT) [20] for synthetic and natural images at different levels of noise. See [26, 27, 28] for more details. Figure 4 shows the results for images without noise and with 2% of the overall image pixels or equivalently 118% of all contour points as randomly distributed noise. The same images have been processed by HT and RHT with 256×256 accumulator arrays that are thus equivalent to the image resolution. Table 1 summarizes the results of a profiling analysis of the core HT, RHT and MIN PFS algorithms on a SUN UltraSparc WS. All results are normalized to 3.57 seconds. The lower part of the Table indicates the (subjective) result quality.

As confirmed by all our experiments [26, 27, 28], the MIN PFS approach provides the higher quality results for all noise conditions. Moreover, its computation time requirements are much lower with respect to HT for low levels of noise and comparable for higher noise levels, while yielding much higher quality results. Notice that HT and RHT fail to provide any result for, respectively, high and medium level of noise. It has to be pointed out that the above comparison would turn out to be much more favorable for larger image sizes. For instance, considering the same image but at a double resolution (512×512 pixels instead of 256×256) HT memory requirements and processing time would increase by a factor of 4, while the time and space complexity of our algorithm would remain the same.

Another example that demonstrates the very good performance and robustness versus noise of our approach is reported in Fig. 5. The image contains 500 points generated by adding a Gaussian noise of $\sigma = 10$ pixels to two original segments. For various noise levels, it always recovers the two original segments within 3.65 seconds of processing time. As shown in Fig. 5, RHT never provides a correct result and 65% of the segments determined by the HT (in 6.25 seconds) are not correctly grouped or do not have the correct parameters values.

Noise % (*)	0	1	2	5	10
(**)	0	59	118	295	590
HT	1	1.16	1.29	1.77	4.79
RHT	0.08	0.13	0.14	0.46	3.30
MIN-PFS	0.06	0.49	0.93	2.19	11.2
HT	++	++	++	-	-
RHT	++	++	+	-	/
MIN-PFS	++	++	++	+	+

Table 1 : Comparison of normalized CPU times (in seconds) and quality of results obtained with HT, RHT and MIN PFS-based algorithms for the image of Figure 4 at different levels of noise. (*) The noise % refers to the overall image points (256×256). (**) The noise % refers to the total contour points. $[++]$ all the information is correctly detected, $[+]$ all the information can be extracted by a simple postprocessing, $[-]$ information is missing, $[/]$ no results can be obtained.

A third typical example is reported in Figure 6. The images under consideration contain 10 randomly distributed segments with additional speckle noise corresponding to 50% of the line points. The MIN PFS-based approach always provides the correct results. Since in average each subsystem corresponding to a line segment only contains $3-5\%$ of the total number of contour points, no method based on robust regression techniques can yield any useful result. The breakdown point of robust regression techniques is far above such values.

5.2 Piecewise linear modeling of time series

Nonlinear signal models have been the object of an increasing interest over the past few years. A number of applications can be found in different fields such as, for instance, economic system modeling and biomedical signal analysis. The idea of breaking a global linear model into a number of submodels, gave rise to the so-called threshold autoregressive (TAR) models [38, 39], in which the choice of the model at time t is based on the comparison of the signal value at time $t-1$ with pre-defined thresholds. This scheme allows one to model and reproduce phenomena such as jumps and limit cycles, but its inherent limitation derives from the selection of the thresholds, for which no general method exists. The principle, called piecewise linear modeling [40], consists of partitioning the state space into a certain number of regions and estimating a linear autoregressive submodel for each one of them. A piecewise linear autoregressive model can be described as follows [40]:

$$y_t = \sum_{j=1}^{n} x_j(Y_t)\, y_{t-j} + u_t \tag{16}$$

where the coefficients $x_j(Y_t)$, $1 \leq j \leq n$, at time t depend on the position of the vector $Y_t = (y_{t-1}, \dots, y_{t-n})$ with respect to a given partition of the state space $\mathbf{R}^n$, and $\{u_t\}$ is an i.i.d. sequence. Note that here the y's are the observations and the coefficients x_j's as well as the partition of the state space have to be estimated.

Although piecewise autoregressive models are more attractive than TAR ones, the number of submodels must be selected *a priori* and an appropriate state-space partition needs to be determined. In [40] a two-stage strategy is suggested. First one determines a state-space decomposition using a Kohonen feature map [19] and then one estimates the parameters of each submodel using robust regression techniques. The idea behind such a unsupervised clustering phase is to avoid guessing the number of submodels in advance. However, besides the delicate convergence issue, it is unclear how a state-space partition can be actually derived from a given feature map. Even more importantly, the clustering process does not take into account the type of submodels to be used.

According to our general approach, we consider the problem of estimating such piecewise linear models in terms of MIN PFS. Clearly, any sequence of observations $\{y_1, \dots, y_L\}$ which cannot be modeled (approximated) by a simple autoregressive model, leads to an overdetermined infeasible system $A\mathbf{x} = \mathbf{b}$ where

$$\mathbf{A} = \begin{bmatrix} y_n & y_{n-1} & \cdots & y_1 \\ y_{n+1} & y_n & \cdots & y_2 \\ \vdots & \vdots & \ddots & \vdots \\ y_{L-1} & y_{L-2} & \cdots & y_{L-n} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ \vdots \\ y_L \end{bmatrix} \tag{17}$$

and n is the prediction order.

Given any partition into feasible subsystems, each feasible subsystem defines a group of vectors in state space and the corresponding solution indicates the parameter values of the corresponding submodel. To deal with noise we introduce pairs of complementary inequalities such as those of Equation (2).

5.2.1 Some typical results

A number of experiments have been carried out with various time series which have been extensively used as benchmark in the literature, see [41]. We describe here two typical examples that illustrates some interesting features of our approach.

In the first example a time series admitting a true piecewise linear state-space representation is considered and results are compared with those from a classical Vector Quantization approach [40]. In these experiments 1000 samples $\{y_t\}_{1 \leq t \leq 1000}$ have been generated from a TAR model described by Equation (16), with $n = 2$ and $\{u_t\}$ an i.i.d. Gaussian noise with variance $\sigma^2 = 0.01$. To have a true piecewise linear model, the coefficients $x_1(Y_t)$ and $x_2(Y_t)$ are defined as follows. Given the three vectors $C_1 = (-1\ 0)$, $C_2 = (0\ 0)$ and $C_3 = (0\ 1)$, the coefficient vector $(x_1(Y_t)\ x_2(Y_t))$ is taken as $(-0.9\ 0.81)$, $(0.1\ 2.5)$ or $(0.8\ 0.1)$ depending on whether $Y_t = (y_{t-1}\ y_{t-2})$ is closest, respectively, to C_1, C_2 or C_3. Thus, as shown in

Figure 7 (b), the state space is partitioned into three regions delimited by two vertical lines crossing the horizontal axis at -0.5 and 0.5

According to [40], a classical learning vector quantization (LVQ) [19] has been applied to the state vector set $\{Y_t\}_{3\leq t\leq 1000}$ in order to obtain three centers representative of its spatial distribution. It has to be emphasized that we here assume that the correct number of submodels (centers) is known a priori while this is generally not the case. LVQ is a stochastic procedure, so 100 Monte Carlo runs which constantly produced three centers with approximate coordinates $(-0.98\ 0.10)$, $(-0.94\ -1.17)$, $(-0.11\ -0.88)$ have been performed. The corresponding partition of the state vectors Y_t's, obtained by grouping those which are closest in Euclidian norm to each of these centers, is displayed in Figure 7 a). This partition differs considerably from the one of the generating TAR model, see Figure 7 b). Since the resulting groups include state vectors corresponding to differents original submodels, estimation of the submodel coefficients based on the given groups is bound to give poor or even meaningless results.

As for instance in [10], when complementary inequalities such as (2) are considered to account for noisy data, an appropriate maximum noise tolerable error $\varepsilon > 0$ needs to be selected. Intuitively, the value of ε should correspond to a trade-off between *model accuracy* and *model complexity*. For too large a value, very large subsystems of equations are considered as feasible. In other words, the overall model is simpler since it contains very few linear submodels but the submodels do not fit well the corresponding data points. Conversely, too small a value of ε leads to a very large number of subsystems, i.e., too many submodels which do not capture the actual underlying structure. For data coming from an actual piecewise linear model and an amplitude limited additive noise (e.g. quantization noise) a value of ε slightly higher than the maximum noise amplitude should be appropriate. Although it may seem that *a priori* information on the underlying model structure and noise characteristics is needed, it suffices to run the method for a wide enough range of values of ε. Given the low computational requirements of our MIN PFS-based method, the curves expressing the number of subsystems (submodels) and the average quadratic error as a function of ε can be sketched even for very large data sets. Typically, when ε increases starting from a very small value, the number of feasible subsystems (linear submodels) first sharply decreases and then remains almost constant on a wide range of values. An appropriate value of ε should clearly be one close to the knee of this curve.

To find a good partition for the time series arising from the above TAR model, the algorithm is run for ε ranging from 0.1 to 5. Results are shown in Figure 8. For values of ε below 0.7 the average quadratic error is small but the large number of subsystems (submodels) clearly indicates overfitting of the data. For values of ε between 0.7 and 1.8, the number of selected subsystems remains constant and equal to the actual number of submodels (3), while the average quadratic error grows moderately with ε. For values of ε beyond 4.5, only one linear submodel is detected for the whole dataset, and the average quadratic error becomes approximately constant. This corresponds to the least mean square fit of a single linear model. Since $\varepsilon = 0.7$ coincide with a local minimum of the average quadratic error and a knee of the number of subsystems curve, the best trade-off between model accuracy and model complexity is achieved in the neighborhood of this value.

Figures 9 and 10 illustrate this ε value selection process in more details. On the left we report the results of the greedy strategy and on the right a reassignment of the data points based on the minimum average quadratic error criterion. Comparison between these pairs of plots provides useful insight. In case the value of ε is too large, as in Figure 9 (a) and (b), the two plots differ substantially. The greedy algorithm assigns to the first subsystem all data points which fall within the too loose error tolerance bound specified by ε. By reassigning a posteriori each data point to the extracted subsystem with the minimum average quadratic error, very different results are obtained. But for close-to-optimal values of ε, as in Figure 9 (c) and (d), the two results are very similar. For smaller values of ε one starts to observe an overfitting of the data (see Figure 10 (a) and (b)), clearly detected by an assignment based on minimum average quadratic error which looses the spatial coherence of the state-space partition. As shown in Figure 10 (c) and (d), when ε is further decreased the results are dominated by the underlying noise.

In the second example of nonlinear signal modeling we consider the time series generated from the classical Henon map [7] defined by:

$$x_t = 1 - a x_{t-1}^2 + b x_{t-2}. \tag{18}$$

For parameters values $a = 1.4$ and $b = 0.3$, the dynamics of this system is chaotic and the points with coordinates $[x_{t-2}\ x_{t-1}]$ evolve on a fractal strange attractor, see Figure 12 (a). While in the previous example the underlying dynamics was truly piecewise linear, the goal here is to approximate the nonlinear dynamics described by Equation (18) by a suitable number of linear submodels. In this sense, the modeling error in predicting x_t with x_{t-2}, x_{t-1} comes from the local linear approximation of Equation (18) and not from the dynamical noise as for the TAR system.

We report the results obtained for an experiment on this map with a sequence of 2000 consecutive samples, with a modeling order of the autoregression $n = 2$ and a bias term included. The first task consisted in selecting an appropriate value of the maximum tolerable error ε. Therefore we applied the above procedure and performed a series of trials for ε ranging from 0.02 to 0.3. Figures 11 a) and b) report the curves of the number of subsystems and average quadratic error versus the value of parameter ε. The situation is a less clear-cut than in the TAR model example, which is not surprising in the light of the remark above. Since the system under study is not piecewise linear in essence, there is no threshold value of ε above which the number of subsystems is adequate, and increases sharply below it. Nevertheless, the two curves in Figure 11 indicate that the average quadratic error tends to grow considerably above $\varepsilon = 0.2$ and that the number of subsystems grows exceedingly for ε smaller than 0.1. As a consequence we selected the partition obtained with $\varepsilon = 0.1$.

The resulting partition consists of 6 feasible subsystems. The regions corresponding to the three largest subsystems are shown in Figures 12 (b), (c), and (d). Interestingly, the regions corresponding to each linear submodel are well defined in the state-space and span specific parts of the attractor, which make them valuable for modeling purposes. Depending on the region to which the point $[x_{t-2}\ x_{t-1}]$ belongs, the corresponding set of autoregressive coefficients provides an estimate of x_t within the predefined value of ε. The fact that the regions associated to different subsystems appear as stripes parallel to the horizontal axis, indicates that nonlinearity is present with respect to x_{t-1} only, which is of course confirmed by Equation (18).

6 Concluding remarks

We have proposed a general combinatorial optimization approach to fit piecewise linear models to data. The problem of piecewise linear model estimation is formulated as that of partitioning infeasible linear systems into a minimum number of feasible subsystems. Given the worst-case complexity of this MIN PFS problem and the type of applications we are interested in, we have presented a simple greedy algorithm which provides good approximate solutions in a short amount of computation time. The distinctive feature of our approach is that it enables simultaneous determination of the data partition and estimation of the parameter values of each linear submodel without requiring any a priori assumption on the number of submodels. By varying a single error tolerance parameter, one can guarantee different levels of accuracy.

The potential and performance of our MIN PFS-based method has been demonstrated on two challenging problems from image and signal processing. The experimental results reported here and in [26, 27, 28, 41] indicate that it overcomes the serious limitations of classical alternatives and compares favorably in terms of complexity and performance.

The new approach can be applied to a variety of other problems for which piecewise linear models are valuable. In [5, 25] it has, for instance, been adapted to the problem of detecting and parameterizing 2-D motion of multiple moving objects from a sequence of images based on pre-computed optical flow fields. In [41] it has also been used to identify nonstationary transfer functions.

Since higher degree polynomials can be viewed as linear functions with respect to their coefficients, the approach can also be extended to the estimation of piecewise polynomial models with submodels of bounded degree.

Acknowledgments

The authors would like to thank Vincent Noel and Jean-Marc Vesin for fruitful collaboration, Leslie Trotter Jr. for helpful discussions and Pierre Hansen for pointing out some related references.

Appendix

As discussed in [12, 2] for the thermal perceptron procedure, the choice of the initial temperature T_0 and the annealing schedule can significantly affect the quality of the solutions. Ideally, T_0 should be of the same order of magnitude as the typical values of the violations v^k. Clearly, the average $<\mid v \mid>= \sum_{k=1}^{p} \mid v^k \mid /p$ provides a reasonable estimate of the typical value of the v^k for a given $\mathbf{x}$. Since the $\mathbf{x}^i$'s and therefore the v^k's can vary considerably during the relaxation procedure, T_0 must be updated accordingly. To try to keep T_0 close to $<\mid v \mid>$ and to attenuate oscillations, one starts with $T_0 = \alpha <\mid v \mid>$, where the average is evaluated for the initial guess $\mathbf{x}^0$, and at the beginning of each random cycle one sets $T_0 = \beta T_0 + (1-\beta) <\mid v \mid>$, where the average is for the current $\mathbf{x}^i$. Updating T_0 at the beginning of each random cycle corresponds to rescaling T with respect to the current $<\mid v \mid>$ without affecting the decrease of T given by γ. In practice, $\mathbf{x}^0$ is randomly generated, typical values of α and β are respectively 2 and $3/2$, and if γ is linearly decreased from T_0 to 0 over the C cycles then $\gamma = 1 - \frac{c}{C}$. See [2, 25] for more details.

References

[1] S. Agmon. The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6:382–392, 1954.

[2] E. Amaldi. *From finding maximum feasible subsystems of linear systems to feedforward neural network design*. Phd thesis no. 1272, Department of Mathematics, Swiss Federal Institute of Technology, Lausanne, 1994.

[3] E. Amaldi and R. Hauser. Randomized subgradient methods for the maximum feasible subsystem problem. Technical report, 2001.

[4] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147(1-2):181–210, 1995.

[5] E. Amaldi and M. Mattavelli. A combinatorial optimization approach to extract piecewise linear structure from nonlinear data and an application to optical flow segmentation. Technical Report 97-12, Cornell Computational Optimization project, Cornell University, Ithaca NY, 1997.

[6] E. Amaldi, M. Pfetsch, and L. Trotter Jr. Some structural and algorithmic properties of the maximum feasible subsystem problem. In G. Cornuéjols, R. Burkard, and G. Woeginger, editors, *Proceeding of the 10th Integer Programming and Combinatorial Optimization conference (IPCO'99), Lecture Notes in Computer Science*, volume 1610, pages 45–59. Springer Verlag, 1999.

[7] J. Argyris, G. Faust, and Haase M. *An Exploration of Chaos*. North Holland, 1994.

[8] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

[9] P. S. Bradley and O. L. Mangasarian. k-Plane clustering. *Journal of Global Optimization*, 16, 2000.

[10] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, algorithms and applications*. Oxford University Press, 1997.

[11] J. Chinneck. Fast heuristics for the maximum feasible subsystem problem. *INFORMS J. on Computing*, 2001. To appear.

[12] M. Frean. A “thermal” perceptron learning rule. *Neural Computation*, 4(6):946–957, 1992.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company, San Francisco, 1979.

[14] J. L. Goffin. The relaxation method for solving systems of linear inequalities. *Math. of OR*, 5:388–414, 1980.

[15] R. Greer. *Trees and Hills: Methodology for Maximizing Functions of Systems of Linear Relations*, volume 22 of *Annals of Discrete Mathematics*. Elsevier science publishers B.V., Amsterdam, 1984.

[16] P. Hansen and B. Jaumard. Clustering analysis and mathematical programming. *Mathematical Programming B*, 79:191–216, 1997.

[17] J. Illingworth and J. Kittler. The adaptive hough transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-9(5):690–698, September 1987.

[18] J. Illingworth and J. Kittler. A survey of the Hough Transform. *Comput. Vison Graphics Image Process.*, 44:87–116, 1988.

[19] T. Kohonen. The self organizing map. *Proceedings of the IEEE*, 78:1464–1480, 1990.

[20] P. Kultaken, L. Xu, and E. Oja. A new curve detection method: Randomized Hough Transform. *Pattern Recognition Letters*, 11:331–338, 1995.

[21] P.J. Rousseeuw L. Kaufman. *Finding groups in data*. John Wiley & Sons, Inc, New York, 1990.

[22] V. F. Leavers. Which Hough Transform? *CVGIP: Image Understading*, 58(2):250–264, 1993.

[23] E. K. Lee, R. J. Gallagher, and M. Zaider. Planning implants of radionuclides for the treatment of prostate cancer: An application of mixed integer programming. *Optima*, 61:1–7, 1999.

[24] O. L. Mangasarian. Misclassification minimization. *Journal of Global Optimization*, 5:309–323, 1994.

[25] M. Mattavelli. *Motion analysis and estimation: from ill-posed discrete inverse linear problems to MPEG-2 coding*. Phd thesis no. 1597, Communication Systems Division, Swiss Federal Institute of Technology, Lausanne, 1997.

[26] M. Mattavelli, V. Noel, and E. Amaldi. A new efficient line detection algorithm based on combinatorial optimization techniques. In *Proceedings of the 1998 International Conference on Image Processing (ICIP98)*, Chicago, Ilinois, October 4-7 1998. IEEE Signal Processing Society.

[27] M. Mattavelli, V. Noel, and E. Amaldi. A new approach for fast line detection based on combinatorial optimization. In *Proceedings of ICIAP99, International Conference on Image Analysis and Processing*, pages 168–173, Venice, Italy, September 27-29 1999.

[28] M. Mattavelli, V. Noel, and E. Amaldi. Fast line detection algorithms based on combinatorial optimization. In *Proceedings of the 4th International Workshop on Visual Form (IWVF4), Capri, Italy*, Lecture Notes in Computer Science. Springer-Verlag, 2001. To appear.

[29] M. L. Minsky and S. Papert. *Perceptrons: An introduction to computational Geometry*. MIT Press, Cambridge, MA, 1988. Expanded edition.

[30] T. S. Motzkin and I. J. Schoenberg. The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6:393–404, 1954.

[31] E. Oja, H. Kalviainen, and P. Hirvonen. Houghtool a software package for Hough transform calculation. In *9th Scandinavian Conf. on Image Analysis*, pages 841–848, Uppsala, Sweden, 1995. http://www.lut.fi/dep/tite/XHoughtool/xhoughtool.html.

[32] P. Palmer, M. Petrou, and J Kittler. A Hough Transform algorithm with a 2-D hypothesis testing kernel. *CVGIP*, 58-2(Image Understanding):221–234, 1993.

[33] M. Parker. *A set covering approach to infeasibility analysis of linear programming problems and related issues*. PhD thesis, Dep. of Mathematics, University of Colorado at Denver, 1995.

[34] B. T. Polyak. Random algorithms for solving convex inequalities. In D. Butnariu, Y. Censor, and S. Reich, editors, *Inherently parallel algorithms in feasibility and other applications*, Studies in Computational Mathematics. Elsevier, Amsterdam, 2001.

[35] P.J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, Inc, New York, 1987.

[36] J. Ryan. Transversals of IIS-hypergraphs. *Congressus Numerantium*, 81:17–22, 1991.

[37] J. Telgen. On relaxation methods for systems of linear inequalities. *European Journal of Operational Research*, 9:184–189, 1982.

[38] H. Tong. *Threshold Models in Nonlinear Time Series Analysis*. Springer, New York, 1983.

[39] H. Tong. *Nonlinear Time Series*. Oxford University Press, Oxford, 1990.

[40] J-M. Vesin. A common generalization framework for two classical nonlinear models. In *Proc. Int. Workshop on nonlinear digital signal processing*, pages 6–2.3, Tampere, Finland, 1993.

[41] J-M. Vesin, M. Mattavelli, and E. Amaldi. A new approach to piecewise linear modeling based on a combinatorial optimization formulation. *Under revision for Physica D*.

[42] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough Transform. *Pattern Recognition Letter*, 11 (5):334–344, 1990.

[43] E. Zapata, N. Guil, and J. Villalba. A fast Hough Transform for segment detection. *IEEE Trans. on Image Processing*, 4-11:1541–1548, 1995.

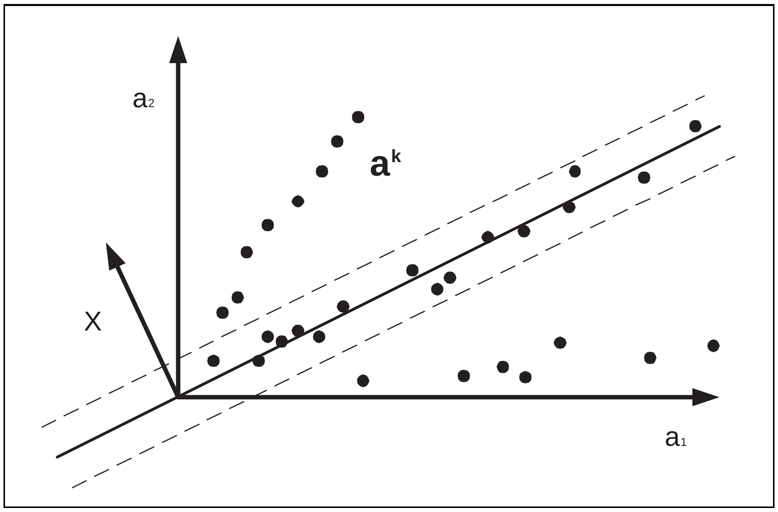


Figure 1. Geometric interpretation in coefficient space of the MIN PFS problem with pairs of complementary inequalities where $b_k = 0$ for $1 \leq k \leq p$ and $\|\mathbf{x}\| = 1$. For arbitrary b_k the hyperslabs do not necessarily contain the origin.

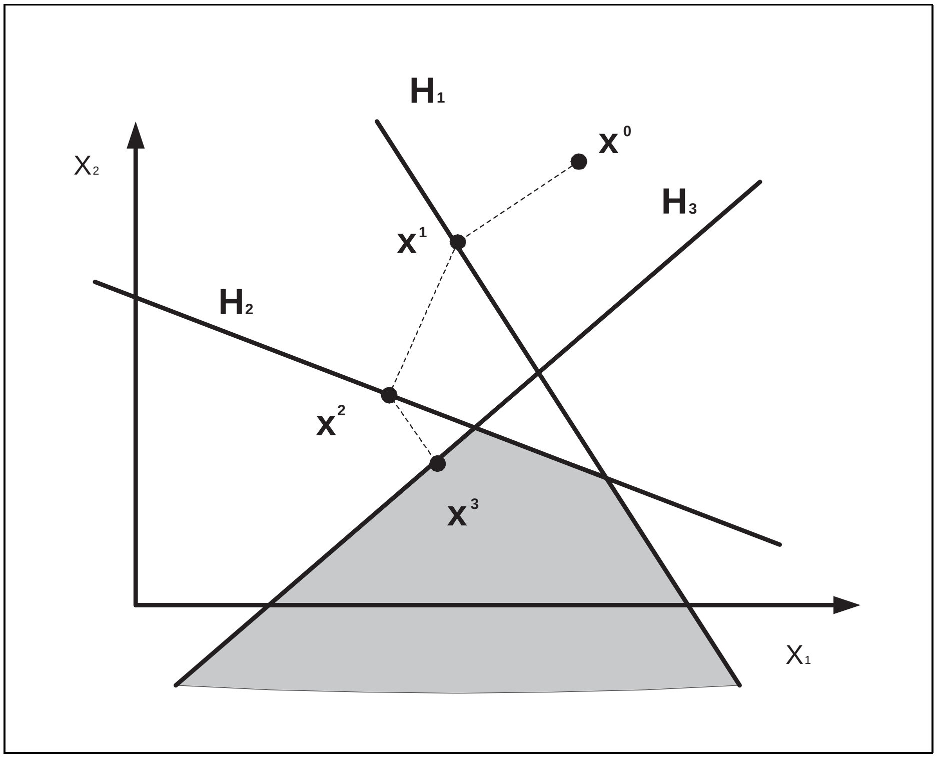


Figure 2. Geometric interpretation in parameter space of the AMS relaxation method for inequalities with $\lambda_i = 1$ and $\|\mathbf{a}^k\| = 1$, $1 \leq k \leq p$.

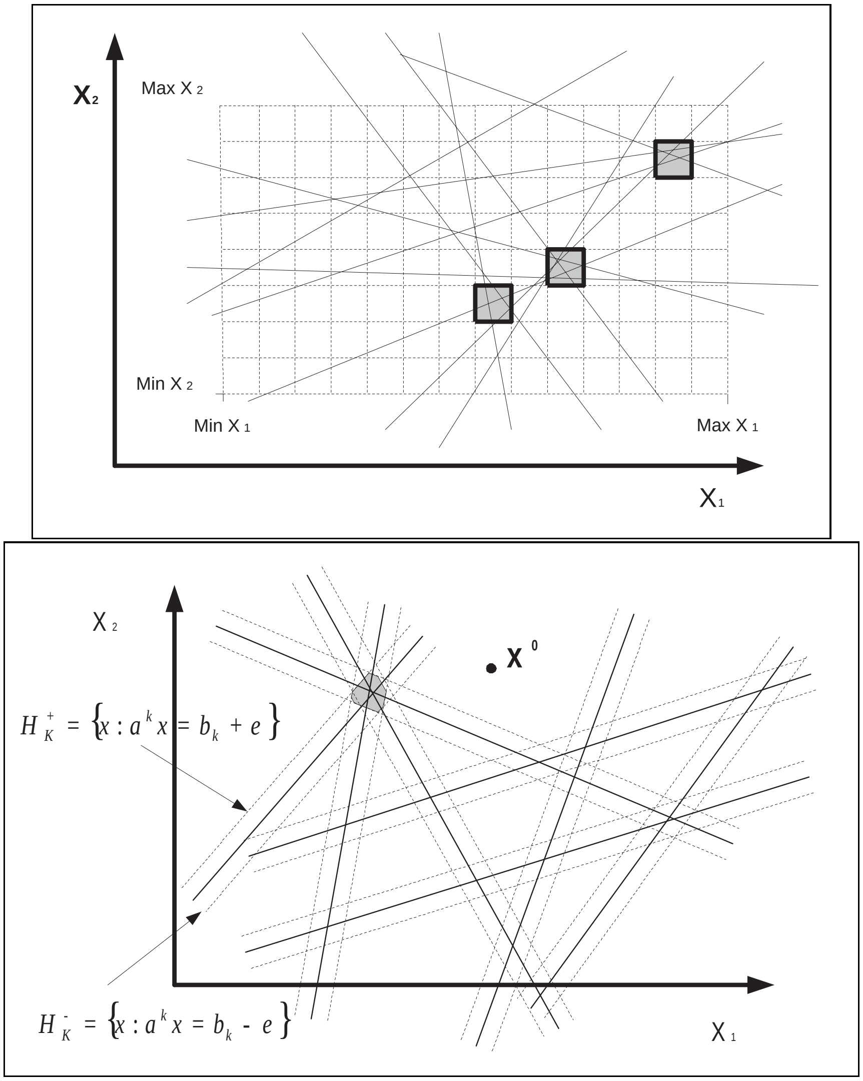

X2
Max X 2
Min X 2
Min X 1
Max X 1
X1
X 2
X 0
$H_K^+ = \{x : a^k x = b_k + e\}$
$H_K^- = \{x : a^k x = b_k - e\}$
X 1

Figure 3. 2-D representation of the HT and MIN PFS-based strategies in parameter space. a) In the standard HT applied to line detection, each data point defines a line in parameter space. Cells are then scanned to find a peak of the counter values, i.e., a cell consistent with a maximum number of lines. b) In our method, for each data point the slab defined by the pair of corresponding complementary inequalities (2) is considered. Starting from an arbitrary $\mathbf{x}^0$, the current estimate is updated according to the thermal AMS procedure (similar to Figure 2) so as to look for an $\mathbf{x}$ contained in the largest number of slabs (see grey region).

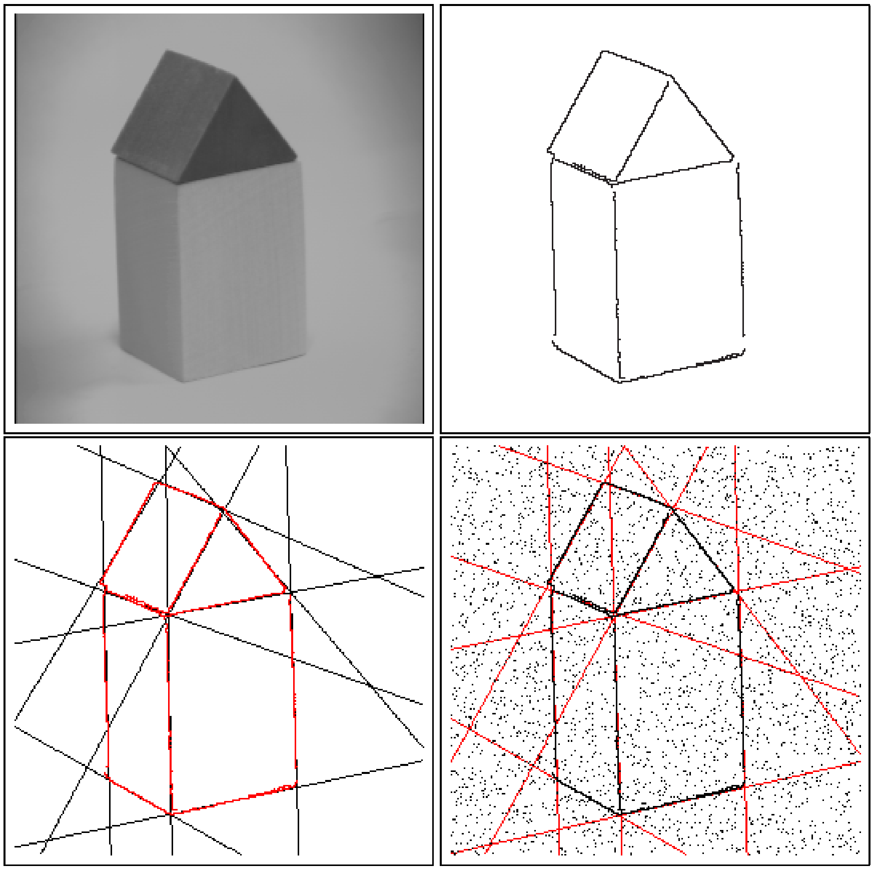


Figure 4. Top left: original grey scale test image. Top right: binary image resulting from basic edge detection. Bottom left: results of MIN PFS-based line detection algorithm. Bottom right: results obtained with 295% of the contour points as random noise (i.e., 5% of the total number of image points)

.

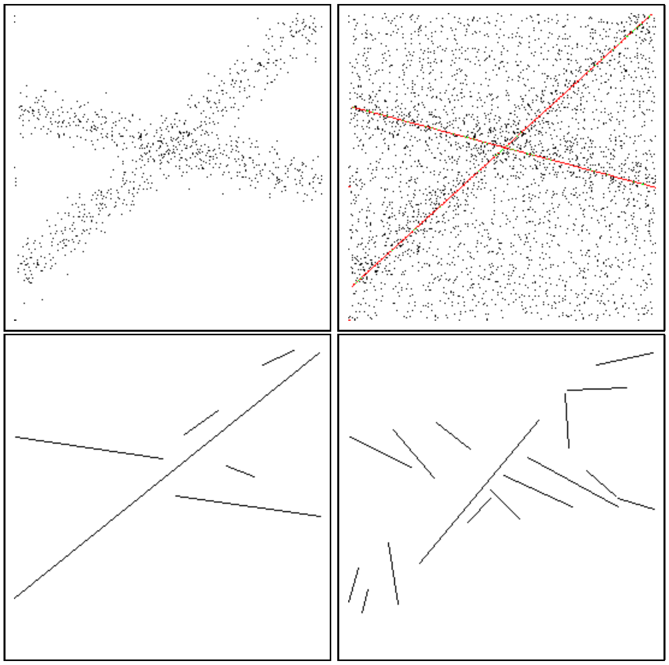


Figure 5. Top left: Points obtained from two line segments by adding a Gaussian noise of $\sigma = 10$ pixels. Top right: MIN PFS-based solution obtained in the presence of 250% of additional randomly distributed noise (w.r.t. original image points). Bottom left to right: results obtained with the HT and RHT.

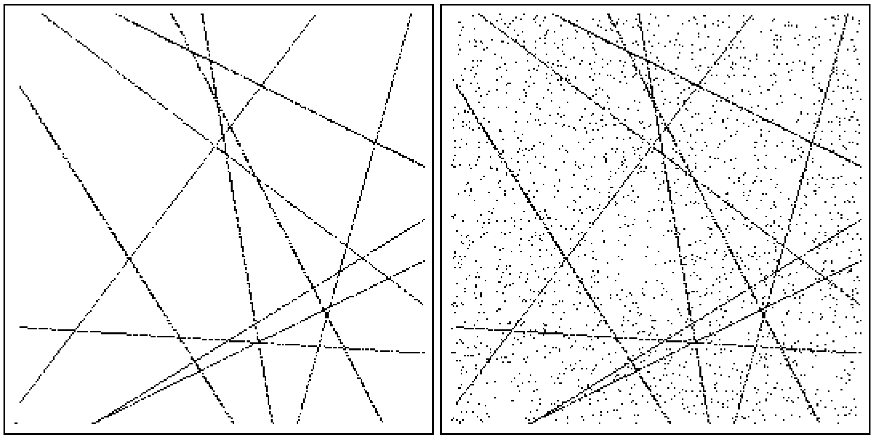
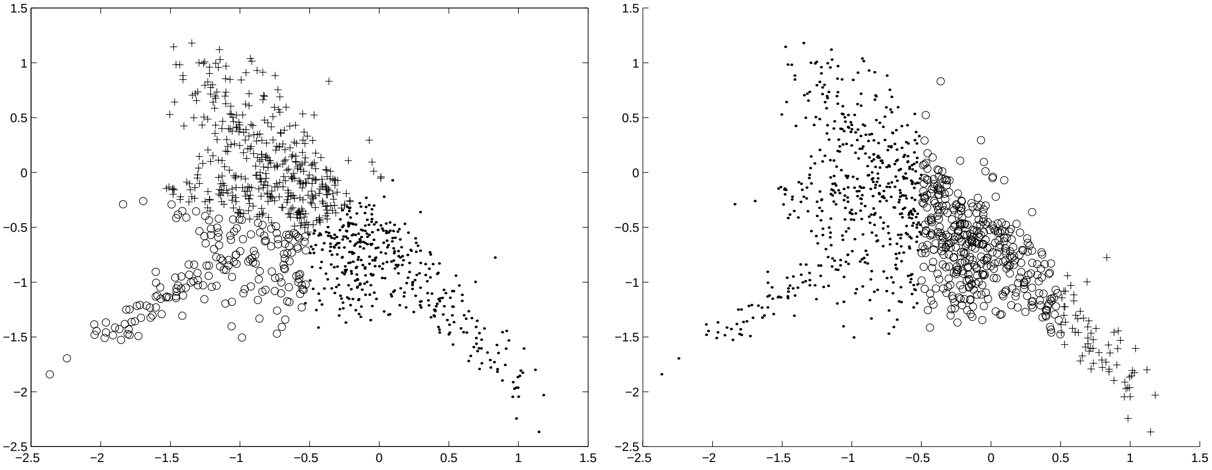


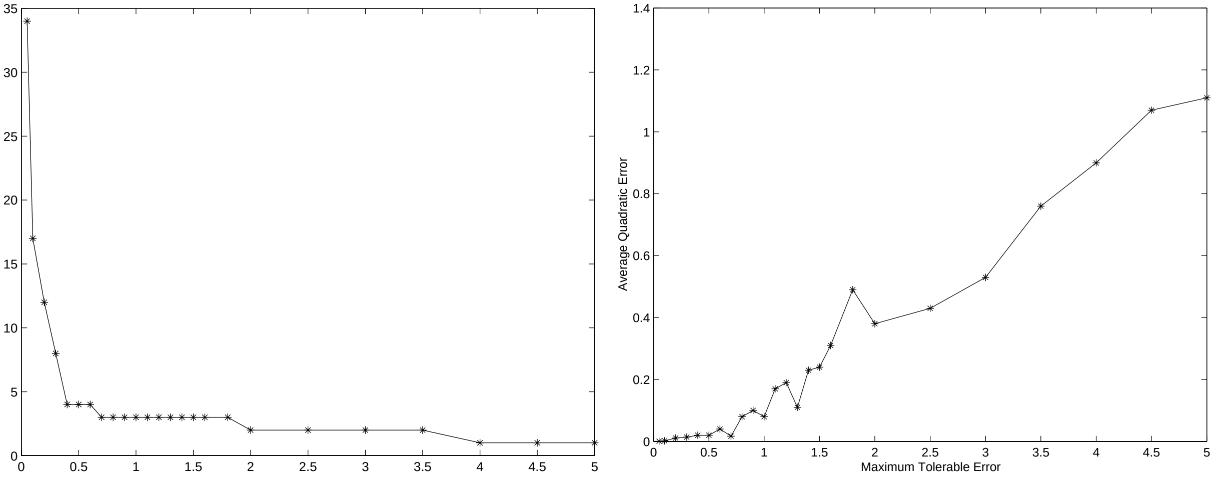
Figure 6. Left: Synthetic image composed of 10 randomly distributed lines. Right: MIN PFS-based solution when 50% are the original image points and 50% are noise (randomly distributed points).



(a)

(b)

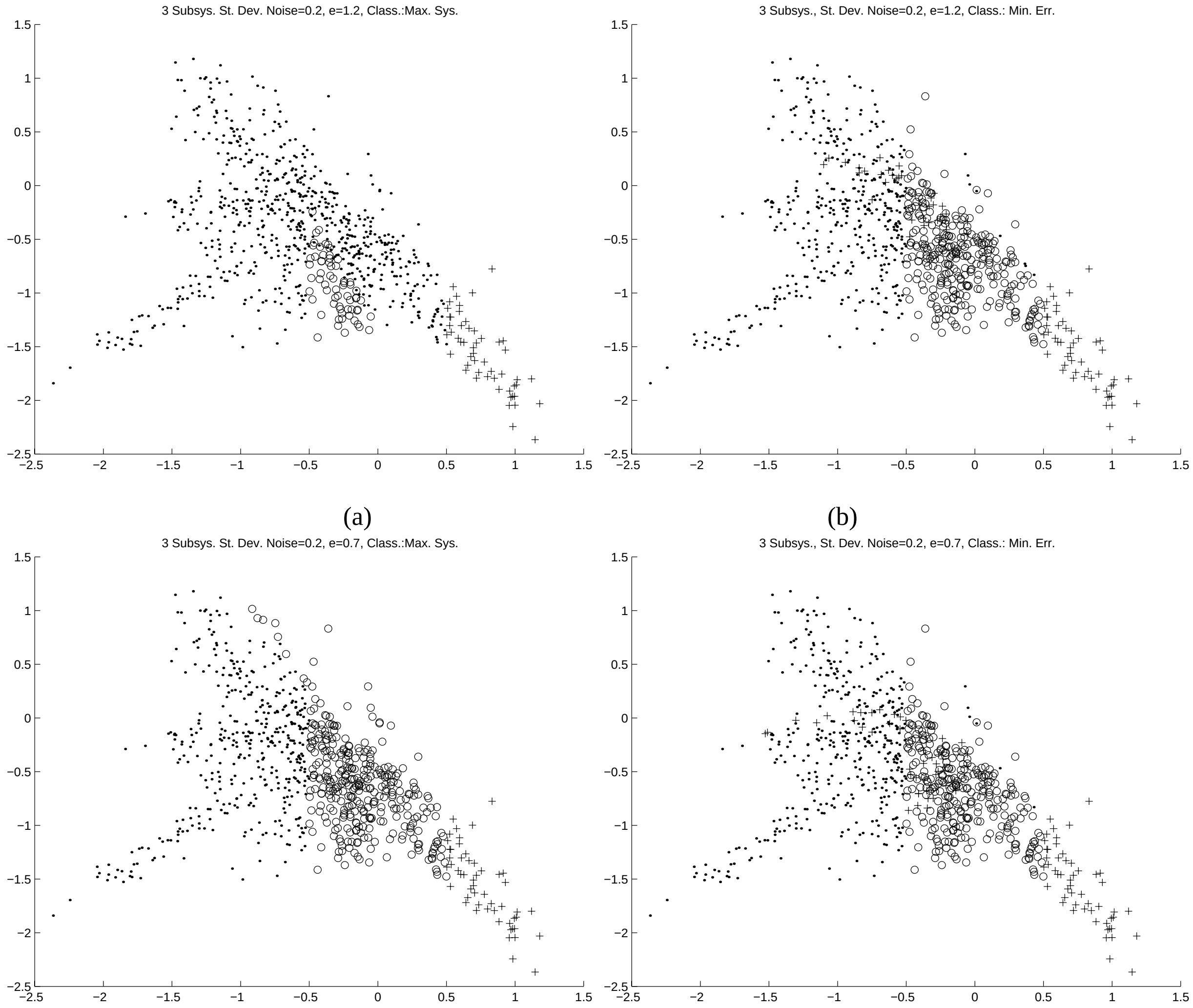
Figure 7. (a) State-vector partition otained using LVQ. The points labeled by (.), (+) and (o) correspond to the three groups induced by the three LVQ centers. (b) Actual partition induced by the TAR model under consideration.



(a)

(b)

Figure 8. TAR model: (a) Number of feasible subsystems and (b) average quadratic error versus the maximum tolerable error (threshold) ϵ.



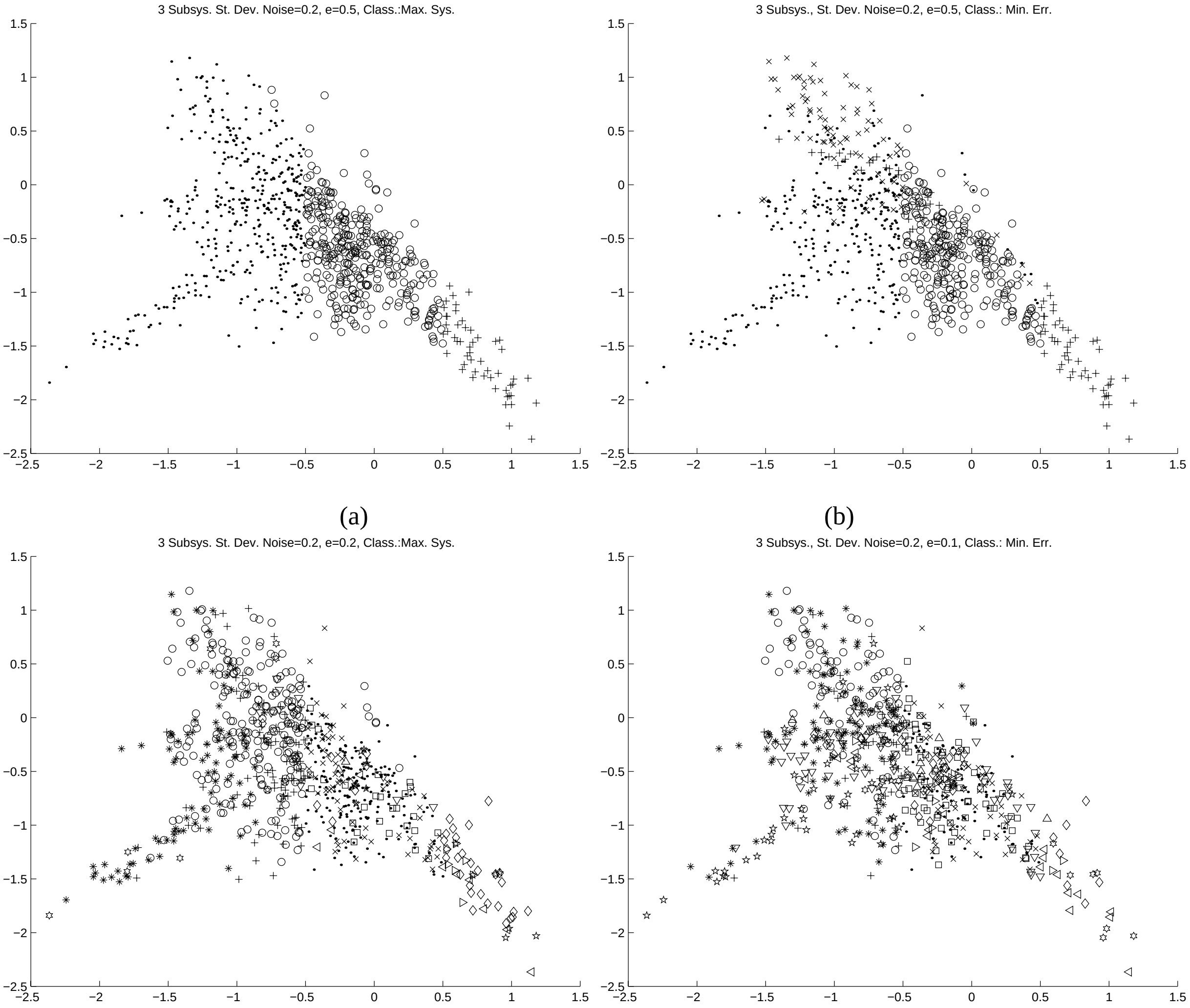
(a)

(b)

(c)

(d)

Figure 9. State-space vector partition provided by MIN PFS method. In (a) and (b) $\epsilon = 0.5$, in (c) and (d) $\epsilon = 0.7$. On the left, the partition is obtained by iteratively extracting the feasible subsystems of close-to-maximum size and by assigning at each iteration the selected state vectors to the corresponding linear submodel. On the right, the final partition is obtained by reassigning at the end of the MIN PFS greedy algorithm each state vector to the submodel providing the smallest quadratic error among all extracted subsystem solutions. As shown in (c) and (d) the two plots are very similar for an appropriate value of ϵ.



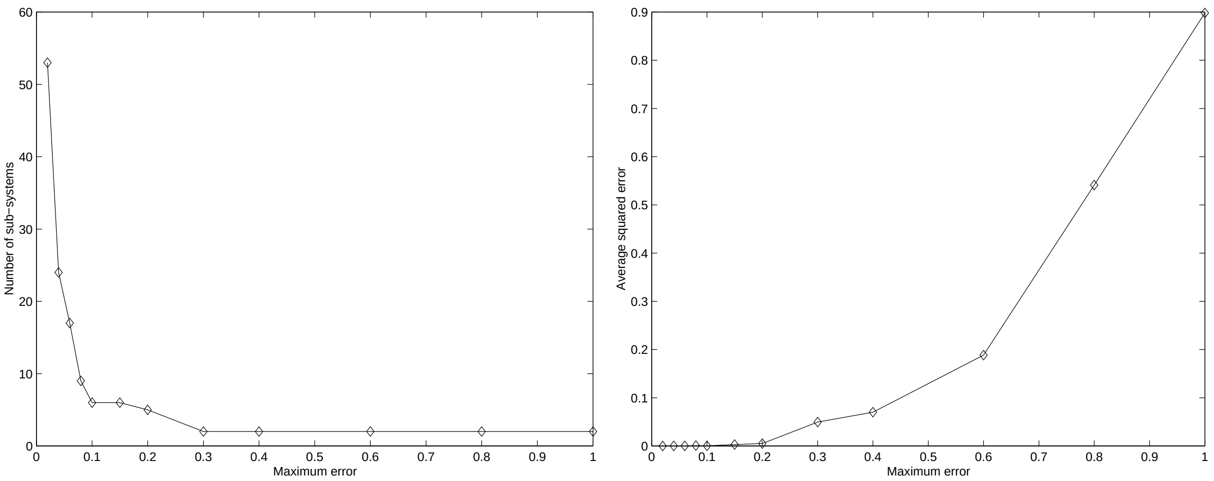
(a)

(b)

(c)

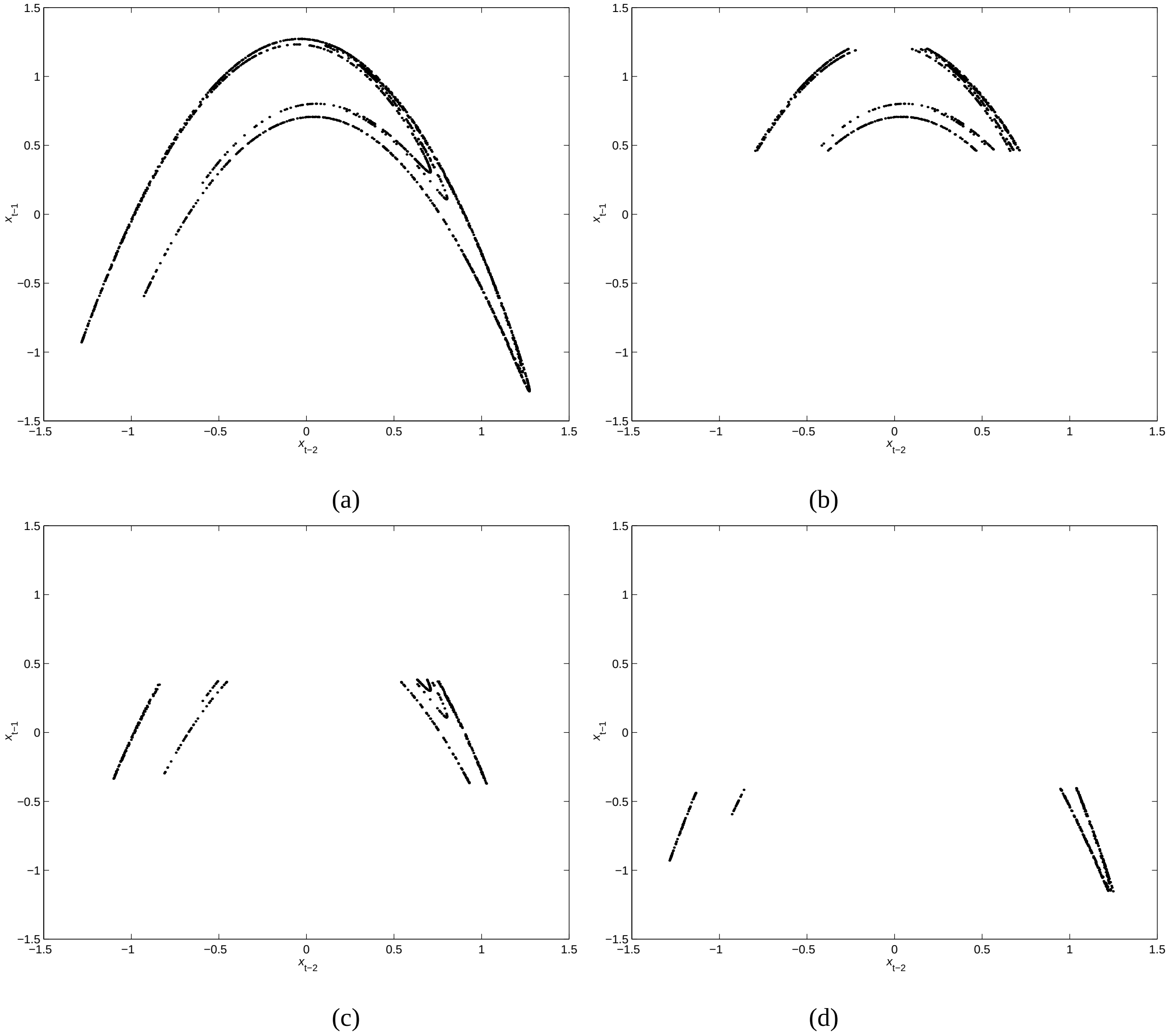
(d)

Figure 10. Figure 9 with lower values of the maximum tolerable error ($\epsilon = 0.5$ in (a) and (b), $\epsilon = 0.1$ in (c) and (d)). Figures (a) and (b): the differences between left and right plots shows the presence of an initial overfitting of the data. For too small values of the maximum tolerable error (Figures (c) and (d)) $\epsilon = 0.1)$ the number of subsystems required in the partition increases and the partition looses its spatial coherence. This situation clearly corresponds to an overfitting of the data.



(a) (b)

Figure 11. Henon map: (a) Number of feasible subsystems and (b) average quadratic error versus the maximum tolerable error ε.



(a)

(b)

(c)

(d)

Figure 12. State-space representation of the Henon time series. Each point at time t is mapped into the point (x_{t-2}, x_{t-1}). $N = 2000$ samples (a) and points corresponding to first (b), second (c), and third (d) largest feasible subsystems.