Fitting Optimal Piecewise Linear Functions Using Genetic Algorithms

Jennifer Pittman and C.A. Murthy

Abstract—Constructing a model for data in $\mathcal{R}^2$ is a common problem in many scientific fields, including pattern recognition, computer vision, and applied mathematics. Often little is known about the process which generated the data or its statistical properties. For example, in fitting a piecewise linear model, the number of pieces, as well as the knot locations, may be unknown. Hence, the method used to build the statistical model should have few assumptions, yet, still provide a model that is optimal in some sense. Such methods can be designed through the use of genetic algorithms. In this paper, we examine the use of genetic algorithms to fit piecewise linear functions to data in $\mathcal{R}^2$. The number of pieces, the location of the knots, and the underlying distribution of the data are assumed to be unknown. We discuss existing methods which attempt to solve this problem and introduce a new method which employs genetic algorithms to optimize the number and location of the pieces. Experimental results are presented which demonstrate the performance of our method and compare it to the performance of several existing methods. We conclude that our method represents a valuable tool for fitting both robust and nonrobust piecewise linear functions.

Index Terms—Genetic algorithms, optimization, statistical data analysis, splines.

1 Introduction

One of the purposes of statistical data analysis is to determine a functional relationship between some input and output variables given a dataset of noisy observed values. The dataset, denoted as $(\mathbf{x}, \mathbf{y})$, is assumed to be a number of realizations of some underlying process combined with random noise, i.e., $\mathbf{y} = f(\mathbf{x}) + \epsilon$, where ϵ has mean zero. Often the function $f(\mathbf{x})$ is assumed to meet certain mathematical requirements, such as continuity or differentiability, and to be of a certain form, such as curvilinear or piecewise linear. The problem of determining $f(\mathbf{x})$ given a set of data points and various assumptions is relevant to many application fields including engineering, chemometrics and materials science [1], [2].

The construction of a functional model for $f(\mathbf{x})$, denoted by $\hat{f}(\mathbf{x})$, can involve classical statistical tools such as kernel methods, regression, and splines [3], as well as more recent techniques such as neural networks, radial basis functions, and genetic algorithms [4], [5], [6]. Unfortunately, it is usually the choice of technique, rather than the data or prior process knowledge, that motivates the placement of artificial mathematical restrictions on the final model [7]. It should also be noted that not all techniques can guarantee convergence to a near-optimal $\hat{f}(\mathbf{x})$. In the case of a piecewise linear model, we cannot assume strict differentiability and we wish to optimize the number of pieces as well as their placement in the model. Hence, the optimization power of our technique and its assumptions are critical. For these reasons, we utilize genetic algorithms as our primary model fitting tool.

Genetic algorithms (GAs) are stochastic optimization tools from the field of artificial intelligence [8]. They are capable of finding near-optimal solutions to problems without the usual mathematical model restrictions [7], [9]. In this work, we discuss how GAs were employed to fit piecewise linear models to data sets in $\mathcal{R}^2$, where the optimal number of pieces (i.e., knots), as well as their placement, were unknown. In Section 2, we present the problem and discuss current methods for function approximation. Then, in Section 3, we introduce genetic algorithms and detail how GAs can be used to fit optimal piecewise-linear functions. After outlining the supporting theory in Section 4, several examples are presented with results in Section 5. We finish with a brief conclusion and mention of areas for future research in Section 6.

2 Problem Statement and Current Methodology

We assume a data set

$$D = \left\{ (\mathrm{x}_1, \mathrm{y}_1), (\mathrm{x}_2, \mathrm{y}_2), \ldots, (\mathrm{x}_\mathrm{N}, \mathrm{y}_\mathrm{N}) \right\},\ (\mathrm{x}_i, \mathrm{y}_i) \in \mathcal{R}^2 \quad \forall\, i = 1, \ldots, N,\ 1 \leq N < \infty,$$

where the values (x_i, y_i) are related by an unknown function f such that $\mathrm{y}_i = f(\mathrm{x}_i) + \epsilon_i$, where ϵ_i is a random error with mean zero and constant variance. The problem is to fit a model $\hat{f}$ to the data such that

- $\hat{f}$ is an h^*-piecewise linear function where h^* is an unknown positive integer representing the number of pieces, $1 \leq h^* \leq h_{\max}$.
- The knot locations $(\mathrm{z}_{11}, \mathrm{z}_{12}), \ldots, (\mathrm{z}_{(\mathrm{h}^*+1)1}, \mathrm{z}_{(\mathrm{h}^*+1)2})$ are unknown (the endpoints of $\hat{f}$ are also considered knots).

- J. Pittman is with the National Institute of Statistical Science, P.O. Box 14006 Research Triangle Park, NC 27709. E-mail: pittman@niss.org.
- C.A. Murthy is with the Machine Intelligence Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta 700035, India. E-mail: murthy@isical.ac.in.

Manuscript received 11 Dec. 1997; revised 15 Oct. 1999; accepted 28 Apr. 2000.
Recommended for acceptance by J. Connell.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107782.

- $\hat{f}$ maximizes the quality of the fit over all such h-piecewise linear functions, $1 \leq h \leq h_{\max}$, where the quality of the fit is determined by a specified evaluation function $ev(\hat{f}(\mathrm{x}))$.

We make no assumptions regarding the smoothness of $\hat{f}$ around its knots.

Classical approximation theory suggests methods for fitting piecewise linear functions which involve building models from linear combinations of nonlinear functions [10]. Such linear estimators can be expressed as:

$$\hat{f}(x) = \sum_{i=1}^{n} K_\lambda(x, \mathrm{x}_i)\mathrm{y}_i, \tag{1}$$

where $K_\lambda(x, \mathrm{x}_i)$ is a weighting function which depends on some parameter(s) λ [3]. Examples include kernel estimates, series approximators, and locally weighted regression (which we will not explicitly discuss), splines, and the more recent neural network and radial basis function estimates.

Splines are useful when we want an estimate which meets a quality of fit (fitness) criterion as well as a smoothness criterion. For example, we may estimate y by choosing $\hat{f}$ to minimize

$$n^{-1}\sum_{j=1}^{N}\left(\mathrm{y}_j - f(\mathrm{x}_j)\right)^2 + \lambda\int_a^b f^{(m)}dx \tag{2}$$

for $\lambda > 0$, $m \in \mathcal{Z}^+$, and $a \leq \mathrm{x}_j \leq b \; \forall \; j = 1, \ldots, N$. The solution $\hat{f}$ is called a *smoothing spline* estimate and λ is the *smoothing parameter*. λ determines the tradeoff between goodness-of-fit and smoothness. If, however, our fitness criterion is not of this form, a spline may not be the best type of estimate. It is of note that finding a good estimate for λ can be computationally demanding [11] with generalized cross-validation (GCV) estimates tending to oversmooth the data [12], while m is often based on prior information [3] as opposed to theoretical considerations. Note that cross-validation (CV) [13] is a criterion designed to minimize predictive error; GCV is simply a modified version of cross-validation in which the diagonal elements of the smoothing matrix of the spline are replaced by their average value. Current spline fitting methods include Schwetlick and Schütze's [14] algorithm which optimizes the location and number of "free" knots (but is computationally "too expensive" to implement) and the algorithms of deBoor and Rice [15] and Dierckx [16] which fit least-squares variable knot splines using two different criteria to determine the location of the knots.

A recent development in functional approximation is the use of *neural networks* (NN) and *radial basis functions* (RBFs). Multilayer neural networks are linear function approximators (in the sense of (1)) of the form, e.g.,

$$\hat{f}(x, \mathcal{W}) = \sum_{j=1}^{M} \beta_j g_j(\alpha_j x), \tag{3}$$

where $\{\beta_j, 1 \leq j \leq M\}$, are the weights connecting M hidden units to the output unit, $\{\alpha_j, 1 \leq j \leq M\}$, are the weights connecting the input layer unit to the jth hidden layer unit, and the g_js are the hidden layer activation functions [10]. $\mathcal{W}$ represents the matrix of network weights. Because of the form of $\hat{f}$, splines and kernel estimates are sometimes viewed as special cases of NN models. Another special case of NN models is based on radial basis functions where the approximation may be represented by the equation

$$\hat{f}(\mathrm{x}_k, \mathcal{W}) = w_0 + \sum_{j=1}^{M} w_j \tau^{-p}\phi\big(\|x - c_j\|/\tau\big), \tag{4}$$

where ϕ is the radial basis function, $\{c_j : j = 1, \ldots, M\}$ is the set of RBF centers, τ is a scale parameter, and p is the dimension of the data. Often, ϕ corresponds to a Gaussian density [4], [5]. Note that a radial basis function network (4) (RBFN) is essentially a kernel method for regression. NNs are easily programmed and have been used to solve numerous complicated optimization problems in high dimensional spaces. However, there is no method for finding the best network architecture for a given problem and NNs do have a tendency to overfit or overparameterize the data [17]. RBFNs have been shown to outperform multilayer perceptrons (MLPs) [5] even though the choice of centers and the curse of dimensionality can make implementation difficult.

Specific algorithms for variable knot piecewise constant/linear modeling can be found in the numerical analysis literature, e.g., Baines [19], Tourigny and Baines [20], and Loach and Wathen [21]. Baines' algorithm aims for optimal discontinuous $\mathcal{L}_2$ fits to continuous functions; by concentrating on discontinuous fits, it effectively linearizes the problem at hand. Tourigny and Baines [20] proved that under certain assumptions, Baines' algorithm generates a mesh sequence which converges to a locally optimal mesh. Although the adjustment necessary to ensure a continuous fit forces the abandonment of an optimal fit, and the algorithm can converge to a local optimum, the algorithm does have a nice extension to the two-variable case.

Loach and Wathen [21], on the other hand, approach the given problem by utilizing results given in Chui et al. [22] and Barrow et al. [18] to design a hybrid algorithm for computing the best $\mathcal{L}_2$ linear spline approximation to a given continuous function with a fixed number of free knots. Given its sensitivity to the choice of initial reference, the algorithm utilizes dynamic programming to generate promising starting values. It will also, in most cases, return a solution with distinct, ordered knot points.

Variable knot models have also been fit using Bayesian estimation as opposed to GA optimization [23], with promising results, although GAs can guarantee convergence to an optimal solution for a sufficient number of iterations [24].

A discussion of the advantages and disadvantages of the above methods led us to consider the possibility of using genetic algorithms (GAs) as a data fitting tool. GAs only need a set of parameters, a way to calculate a response, and an evaluation function (i.e., a measure of quality of fit) to construct a model for a data set. The artificial requirements of differentiability and smoothness imposed on the functional form of the model by the above methods can be discarded [7] and criteria more robust to outliers in comparison to $\mathcal{L}_2$ error can be considered. In the case of piecewise linear models, the power of GA optimization allows variable knot placement as well as a variable number of knots. Related works include those of Karr et al. [2], [6], who have applied GAs to the problem of least-squares (LS) and least-median-squares (LMS) curve fitting where the specific form of the curve is known (e.g., a polynomial of degree 2) and the data is noiseless, and Vankeerberghen et al. [25] who have used GAs to obtain the parameters for specific, known laboratory system models (e.g., a hyperbolic model) which are nonlinear in the parameters. We intend to utilize GAs for fitting both robust and nonrobust optimal piecewise linear functions to data in $\mathcal{R}^2$.

3 Genetic Algorithms

Genetic algorithms are stochastic search methods which provide a near optimal solution to the evaluation function of an optimization problem [7]. They can be used to search complex, multimodal surfaces via steps which have been designed to mimic the processes of natural genetic systems. Their effectiveness has been demonstrated in solving various problems from scientific fields such as scheduling, classifier systems, and pattern recognition [9].

Each possible solution is encoded as a string or chromosome; a set of such chromosomes is called a *population*. An evaluation (*fitness*) function provides a mapping from the chromosome space to the solution space. GAs start with an initial population of a fixed number of randomly generated strings. At each iteration, three basic operations—selection, crossover, and mutation—are applied over the current population to yield a new population of strings. This cycle is repeated until some termination criterion is met, at which time the best string achieved is generally taken as the solution to the optimization problem.

This basic framework can be modified to specifically address the problem of interest by, e.g., the choice of population or the addition of other operators. An example of one such algorithm is the variable length genetic algorithm (VLGA) designed by Bandyopadhyay et al. [26] in which the length of each string is allowed to vary depending on the number of parameters in the given solution. In this way, the VLGA is capable of handling a solution space containing solutions of varying dimensions. Although a VLGA will not be implemented here due to considerations of computational expense, a VLGA could be used as an optimization tool in solving the problem at hand.

3.1 Remarks

3.1.1 Convergence

Bhandari et. al. [24] have proven theoretically that an elitist genetic algorithm (fixed length strings) will converge to the optimal string as the number of iterations goes to infinity. This convergence is independent of the choice of values for the algorithm parameters (M, p_s, p_c, p_m, etc.), although these parameter values do influence the rate of convergence. There is no theory to indicate the number of iterations necessary for convergence. Two popular heuristic stopping rules are: 1) Execute the process for a fixed number of iterations and report the best string found as the solution, or 2) Execute the process until the fitness value does not show adequate improvement over a fixed number of iterations, and report the best string found as the solution.

3.1.2 Flexibility

GAs can be applied to a wide range of optimization problems with little adjustment—in most cases, only the fitness function needs to be redefined. Thus, it is possible to use the same basic algorithm to, for example, fit lines satisfying different optimization criterions to a given dataset.

4 Theory of Piecewise Linear Fitting in $\mathcal{R}^2$

4.1 Mathematical Formulation

Our stated goal is to fit piecewise linear functions to datasets in $\mathcal{R}^2$. Generally speaking, if we think of a given dataset as a realization of some random variable, we would like our set of lines to represent the center of the density of that random variable.

Let us assume that we have h^* lines, where $1 \le h^* \le h_{\max}$, $h_{\max}$ is a positive integer. For each j, $j = 1, \ldots, h^*$, let the equation of the jth line be

$$x \cos\theta_j + y \sin\theta_j = d_j$$

for some $\theta_j \in (0, \pi]$ and $d_j \in \mathcal{R}$. Assume that the jth and $(j+1)$th lines intersect at the point $(\mathrm{z}_{(\mathrm{j}+1)1}, \mathrm{z}_{(\mathrm{j}+1)2})$. We denote the first knot as $(\mathrm{z}_{11}, \mathrm{z}_{12})$ and the last knot as $(\mathrm{z}_{(\mathrm{h}^*+1)1}, \mathrm{z}_{(\mathrm{h}^*+1)2})$.

Let $\epsilon_0 > 0$ and let the set B_j be expressed as:

$$B_j = \left\{(x, y) : y \in \left[\frac{d_j - x\cos\theta_j}{\sin\theta_j} - \epsilon_0, \frac{d_j - x\cos\theta_j}{\sin\theta_j} + \epsilon_0\right]; \right.$$
$$\left. j = 1, \ldots, h^*, x \ \in \left[\mathrm{z}_{\mathrm{j}1}, \mathrm{z}_{(\mathrm{j}+1)1}\right]\right\}.$$

We denote $\bigcup_{j=1}^{h^*} B_j$ as B. The probability density function of (x, y) on B is denoted by $\alpha : B \to [0, \infty)$, where

$$\alpha(x, y) = \begin{cases} \alpha_j(x, y) & x \in [\mathrm{z}_{\mathrm{j}1}, \mathrm{z}_{(\mathrm{j}+1)1}], \ \text{for } j = 1, \ldots, h^* \\ 0 & \text{otherwise.} \end{cases}$$

We define our probability measure P as $P(A) = \int_A \alpha(x, y)\, dx, dy$ for all Borel $A \subseteq \mathcal{B}(B)$, the Borel σ-field of B.

Let $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ be independent, identically distributed random vectors with density α, i.e., there exists a probability space $(\Omega, \mathcal{A}, Q)$ such that

$$(x_i, y_i) : (\Omega, \mathcal{A}, Q) \to (B, \mathcal{B}(B), P), \ i = 1, \ldots, N,$$

where $Q(C) = P((x_i, y_i)(C)) \ \forall \ C \in \mathcal{A}$.

We also assume that

1. For each $j = 1, \ldots, h^*$,

$$\alpha_j(x, y) = \begin{cases} \alpha_j\Big(x, \frac{2(d_j - x\cos\theta_j)}{\sin\theta_j} - y\Big) & x \in [\mathrm{z}_{\mathrm{j}1}, \mathrm{z}_{(\mathrm{j}+1)1}] \\ 0 & x \notin [\mathrm{z}_{\mathrm{j}1}, \mathrm{z}_{(\mathrm{j}+1)1}] \end{cases} \tag{5}$$

   (i.e., α_j is symmetric about the line

$$x \cos\theta_j + y \sin\theta_j = d_j.)$$

2. $\alpha_j(x, y) > 0 \ \forall \ (x, y) \in B_j \ \forall \ j$
3. $\alpha(x, y)$ and $\alpha_j(x, y)$, $j = 1, \ldots, h^*$, are continuous.
4. $\int_B \alpha(x, y)\, dx\, dy = 1$.

In the above mathematical formulation, the primary assumption is the assumption of symmetry of the underlying density around a piecewise linear function (Assumption 1). The other stated assumptions are common properties of a continuous probability density function.

4.2 Solution Space

In order to represent our solution space, suppose we are given a data set $D=(\mathbf{x},\mathbf{y})=\{(\mathrm{x}_1,\mathrm{y}_1),(\mathrm{x}_2,\mathrm{y}_2),\ldots,(\mathrm{x}_\mathrm{N},\mathrm{y}_\mathrm{N})\}$, which is a realization of the random vectors (x_i,y_i), i.e., $x_i(\omega)=\mathrm{x}_i, y_i(\omega)=\mathrm{y}_i$ for some $\omega\in\Omega,\ i=1,\ldots,N$. Let's, define

$$\begin{aligned}
\mathrm{x}_{(1)}&=\min\{\mathrm{x}_i;\ i=1,\ldots,N\},\\
\mathrm{x}_{(N)}&=\max\{\mathrm{x}_i;\ i=1,\ldots,N\},\\
\mathrm{y}_{(1)}&=\min\{\mathrm{y}_i;\ i=1,\ldots,N\},\\
\mathrm{y}_{(N)}&=\max\{\mathrm{y}_i;\ i=1,\ldots,N\}.
\end{aligned}$$

GAs try to find an optimal solution over a finite solution space. Thus, the solution space, i.e., the collection of all h-piecewise linear functions, where $h\in\mathcal{H},\ \mathcal{H}=\{1,\ldots,h_{\max}\}$, must be discretized. To formulate the problem mathematically, we proceed in the following way:

Let L_j denote the straight line

$$x\cos\theta_j+y\sin\theta_j=d_j,$$

where j belongs to an index set. Let $L_{j,\cdot,h}$ represent an h-piecewise linear function, i.e.,

$$L_{j,\cdot,h}=\{L_{j,1,h},L_{j,2,h},\ldots,L_{j,h,h}\},$$

where $L_{j,i,h}$ denotes the ith straight line among the set of h straight lines, $i=1,\ldots,h$, and each $L_{j,i,h}$ satisfies the following properties:

1. $L_{j,i,h}$ represents the straight line
$$x\cos\theta_{j,i,h}+y\sin\theta_{j,i,h}=d_{j,i,h},$$
where $\theta_{j,i,h}$ $(0<\theta_{j,i,h}\le\pi)$ is the polar angle formed when the polar axis is taken as the y-axis and the origin is taken as the intersection point between the y-axis and $L_{j,i,h}$; $d_{j,i,h}$ is the perpendicular distance of the line from the point (0,0).
2. For every $i,\ i=1,\ldots,h-1$, $L_{j,i,h}$ and $L_{j,(i+1),h}$ intersect and the point of intersection is $(\mathrm{z}_{j,(i+1),h,1},\mathrm{z}_{j,(i+1),h,2})$.
3. $$\begin{aligned}
\mathrm{z}_{j,1,h,1}&=\mathrm{x}_{(1)},\\
\mathrm{z}_{j,1,(h+1),1}&=\mathrm{x}_{(N)},\\
\mathrm{z}_{j,i,h,1}&\le\mathrm{z}_{j,(i+1),h,1}\forall\ i=1,\ldots,h.
\end{aligned}$$
4. $L_{j,\cdot,h}=L_{j,i,h}$ if $\mathrm{z}_{j,i,h,1}\le x\le\mathrm{z}_{j,(i+1),h,1},\ i=1,\ldots,h.$

The knot locations

$$(\mathbf{z}_{j,\cdot,h,1},\mathbf{z}_{j,\cdot,h,2})=\big((\mathrm{z}_{\mathrm{j,1,h,1}},\mathrm{z}_{\mathrm{j,1,h,2}}),\ldots,(\mathrm{z}_{\mathrm{j,(h+1),h,1}},\mathrm{z}_{\mathrm{j,(h+1),h,2}})\big)$$

of any $L_{j,\cdot,h},\ 1\le h\le h_{\max}$, are not restricted to the datapoints $\{(\mathrm{x}_1,\mathrm{y}_1),(\mathrm{x}_2,\mathrm{y}_2),\ldots,(\mathrm{x}_\mathrm{N},\mathrm{y}_\mathrm{N})\}$.

For each $h,\ 1\le h\le h_{\max}$, let $\mathcal{L}_h$ represent the class of all h-piecewise linear functions $L_{j,\cdot,h}$ which satisfy the above properties. Then, $\bigcup_{h\in\mathcal{H}}\mathcal{L}_h$ is the collection of functions under consideration.

Note that $\bigcup_{h\in\mathcal{H}}\mathcal{L}_h$ is uncountable. In order to obtain a "near optimal" function from this space, we need to discretize $\bigcup_{h\in\mathcal{H}}\mathcal{L}_h$. We can achieve this by restricting the values of θ and d. Let $\mathrm{l_a}$ be the number of bits used to express θ and let $\mathrm{l_d}$ be the number of bits used to express d. We restrict $\theta_{j,i,h}$ to the values $\{\frac{\pi}{2^{\mathrm{l_a}}},\frac{2\pi}{2^{\mathrm{l_a}}},\ldots,\frac{(2^{\mathrm{l_a}}-1)\pi}{2^{\mathrm{l_a}}},\pi\}$. In specifying $d_{j,i,h}$, we utilize the rectangle *rect* formed by the points $(x_{(1)},y_{(1)})$, $(x_{(N)},y_{(1)})$, $(x_{(1)},y_{(N)})$, and $(x_{(N)},y_{(N)})$. Note that *rect* contains the entire given data set. Let *diag* be the length of the diagonal of *rect* and let ℓ_θ be defined as:

$$\ell_\theta=\begin{cases}\mathrm{x}_{(1)}\cos\theta+\mathrm{y}_{(1)}\sin\theta & \text{if } 0<\theta<\pi/2\\ \mathrm{x}_{(N)}\cos\theta+\mathrm{y}_{(1)}\sin\theta & \text{if } \pi/2\le\theta\le\pi.\end{cases}$$

Then, for a given $\theta_{j,i,h}$, $d_{j,i,h}$ may only take values within the set

$$\Big\{d_{j,i,h}=\ell_{\theta_{j,i,h}}+k_{j,i,h}\delta : k_{j,i,h}\in\{0,1,\ldots,2^{\mathrm{l_d}}-1\},\quad \delta=diag/(2^{\mathrm{l_d}}-1)\Big\}.$$

Note that a line with $d=l_\theta$ intersects *rect* at the point $(x_{(1)},y_{(1)})$ if $0<\theta\le\pi/2$ or the point $(x_{(N)},y_{(1)})$ if $\pi/2\le\theta<\pi$ (the parameter $k_{j,i,h}\delta,\ 0\le k_{j,i,h}\delta\le diag$, is sometimes referred to as the *offset* value).

For each $h,\ 1\le h\le h_{\max}$, let $\mathcal{L}_h^0(\Theta,\mathcal{K})$ denote the finite set of functions in $\mathcal{L}_h$ which satisfy these restrictions, where θ has Θ possible values and k has $\mathcal{K}$ possible values (*note that* $\Theta=2^{\mathrm{l_a}}$ *and* $\mathcal{K}=2^{\mathrm{l_d}}$), i.e.,

$$\begin{aligned}
\mathcal{L}_h^0(\Theta,\mathcal{K})=\Big\{&L_{j,\cdot,h}\in\mathcal{L}_h : L_{j,i,h}\text{ is of the form}\\
&x\cos\theta_{j,i,h}+y\sin\theta_{j,i,h}=\ell_{\theta_{j,i,h}}+k_{j,i,h}\delta\ \ \forall\ i=1,\ldots,h,\\
&\theta_{j,i,h}\in\{\frac{\pi}{2^{\mathrm{l_a}}},\frac{2\pi}{2^{\mathrm{l_a}}},\ldots,\frac{(2^{\mathrm{l_a}}-1)\pi}{2^{\mathrm{l_a}}},\pi\},\\
&k_{j,i,h}\in\{0,1,\ldots,2^{\mathrm{l_d}}-1\},\text{ and }\delta=diag/(2^{\mathrm{l_d}}-1)\Big\}.
\end{aligned}$$

Hence, $\{\mathcal{L}_h^0(\Theta,\mathcal{K}):\mathrm{l_a}=1,2,\ldots\}$ and $\{\mathcal{L}_h^0(\Theta,\mathcal{K}):\mathrm{l_d}=1,2,\ldots\}$ each represent an increasing sequence of nested sets. For sake of clarity, we will henceforth specify $L_{j,\cdot,h}$ as $L(\boldsymbol{\theta}_{j,\cdot,h},\boldsymbol{k}_{j,\cdot,h})$ and denote $L_{j,\cdot,h}(x)$ as $L(\boldsymbol{\theta}_{j,\cdot,h},\boldsymbol{k}_{j,\cdot,h})(x)$. Note that $\boldsymbol{\theta}_{j,\cdot,h}$ and $\boldsymbol{k}_{j,\cdot,h}$ represent the vectors $(\theta_{j,1,h},\ldots,\theta_{j,h,h})$ and $(k_{j,1,h},\ldots,k_{j,h,h})$.

For $L(\boldsymbol{\theta}_{j,\cdot,h},\boldsymbol{k}_{j,\cdot,h})\in\mathcal{L}_h^0(\Theta,\mathcal{K})$, we define

$$\begin{aligned}
&E_{\epsilon,L(\boldsymbol{\theta}_{j,\cdot,h},\boldsymbol{k}_{j,\cdot,h}),a,b}=\\
&\Big\{(x,y):y\in\big(L(\boldsymbol{\theta}_{j,\cdot,h},\boldsymbol{k}_{j,\cdot,h})(x)-\epsilon,L(\boldsymbol{\theta}_{j,\cdot,h},\boldsymbol{k}_{j,\cdot,h})(x)+\epsilon\big),\\
&\qquad x\in[a,b]\Big\}.
\end{aligned}$$

for $a<b,\ \epsilon>0$, and let

$$E_{\epsilon,L(\boldsymbol{\theta}_{j,\cdot,h},\boldsymbol{k}_{j,\cdot,h})}=\bigcup_b\bigcup_{a<b}E_{\epsilon,L(\boldsymbol{\theta}_{j,\cdot,h},\boldsymbol{k}_{j,\cdot,h}),a,b}.$$

If we recall the piecewise linear function which lies at the center of the density of (x,y) and we denote it as ϕ, i.e.,

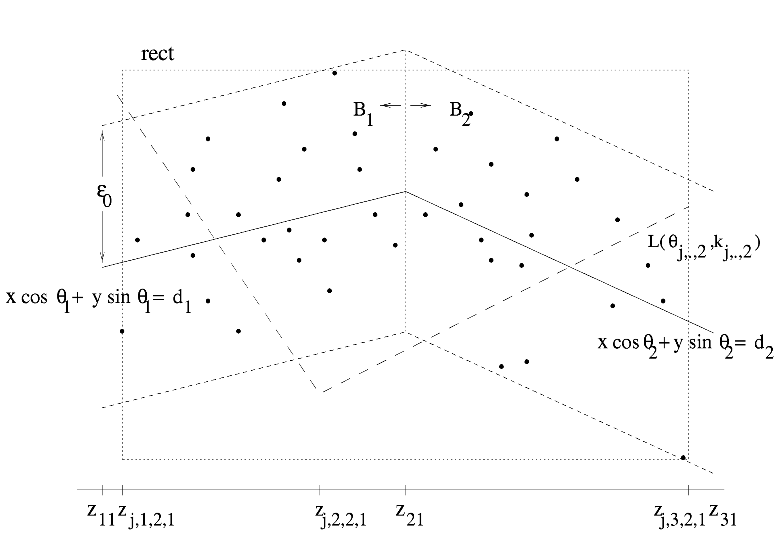

rect
$B_1 \leftarrow\rightarrow B_2$
ϵ_0
$L(\theta_{j,.,2}, k_{j,.,2})$
$x \cos\theta_1 + y \sin\theta_1 = d_1$
$x \cos\theta_2 + y \sin\theta_2 = d_2$
z_{11}
$z_{j,1,2,1}$
$z_{j,2,2,1}$
z_{21}
$z_{j,3,2,1}$
z_{31}

Fig. 1. Support $B = B_1 \bigcup B_2$ with center lines; example function from $\mathcal{L}_2^0(\Theta, \mathcal{K})$.

$$\phi(x) = \begin{cases} \frac{d_j - x\cos\theta_j}{\sin\theta_j} & x \in [\mathrm{z_{j1}}, \mathrm{z_{(j+1)1}}], \quad \text{for j} = 1, \dots, \mathrm{h}^* \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

then, the support B of the density is equivalent to $E_{\epsilon_0, \phi, \mathrm{z_{11}}, \mathrm{z_{(h^*+1)1}}}$.

Corresponding to each $E_{\epsilon, L(\boldsymbol{\theta}_{j,\cdot,h}, \boldsymbol{k}_{j,\cdot,h})}$ we define the set of lines

$$\mathcal{L}_h^{(\epsilon)} = \left\{ L(\boldsymbol{\theta}_{j,\cdot,h}, \boldsymbol{k}_{j,\cdot,h}) : P\Big(E_{\epsilon, L(\boldsymbol{\theta}_{j,\cdot,h}, \boldsymbol{k}_{j,\cdot,h})}\Big) \geq 0.95 \right\},$$

where $P(A) = \int_A \alpha(x, y)\, dx\, dy$ for A Borel, $A \subseteq \mathcal{B}(B)$. Just as $\{\mathcal{L}_h^0(\Theta, \mathcal{K}) : \mathrm{l_a} = 1, 2, \ldots\}$ and $\{\mathcal{L}_h^0(\Theta, \mathcal{K}) : \mathrm{l_d} = 1, 2, \ldots\}$ represent increasing sequences of sets, so does $\{\mathcal{L}_h^{(\epsilon)} : \epsilon = \epsilon_1, \epsilon_2, \ldots; \epsilon_i < \epsilon_{i+1},\ i = 1, 2, \ldots\}$. These definitions involving $E_{\epsilon, L(\boldsymbol{\theta}_{j,\cdot,h}, \boldsymbol{k}_{j,\cdot,h})}$ will be used to define our optimization criterion.

Fig. 1 shows an example dataset where the support $B = B_1 \bigcup B_2$ of the density $\alpha(x, y)$ of (x, y) is centered around a 2-piece linear function. A function from $\mathcal{L}_2^0$ and *rect* is also shown.

4.3 Optimization Criterion

The functional model which is chosen to represent a given dataset is often that which minimizes the sum of the squared errors (i.e., the least-squares function). However, a criterion based on least-squares can yield a solution that is not **robust**—outliers in the dataset can pull the least-squares function away from most of the data points and, hence, away from ϕ [25], [29]. For this reason, we choose not to use least-squares as our optimization criterion. Instead, we note that our concept of a "fitted" function is one which represents the center of a symmetric density function. If $L(\boldsymbol{\theta}_{j,\cdot,h'}, \boldsymbol{k}_{j,\cdot,h'})$ represents our "fitted" function, then, given a dataset D, the majority of the data points in D should fall within the region

$$\left\{ L(\boldsymbol{\theta}_{j,\cdot,h'}, \boldsymbol{k}_{j,\cdot,h'})(x) - \epsilon, L(\boldsymbol{\theta}_{j,\cdot,h'}, \boldsymbol{k}_{j,\cdot,h'})(x) + \epsilon), \right.$$
$$\left. x \in \left[z_{j,1,h',1}, z_{j,(h^*+1),h',1}\right] \right\}$$

for some "small" $\epsilon > 0$. This observation is the basis of our optimization criterion.

Let $\psi_{j,\cdot,h}$ denote an h-piecewise linear function where each piece $\psi_{j,i,h},\ i = 1, \ldots, h$, satisfies the properties listed above for $L_{j,i,h}$. For any $\epsilon > 0$, define

$$\mathcal{I}_{\epsilon, \psi_{j,\cdot,h}}(x, y) = \begin{cases} 1 & \mid y - \psi_{j,\cdot,h}(x) \mid < \epsilon \\ 0 & \text{otherwise.} \end{cases}$$

Our function to be optimized (i.e., fitness function) may then be stated as:

$$ev_{\epsilon,N}(\psi_{j,\cdot,h}) = \sum_{i=1}^{N} \mathcal{I}_{\epsilon, \psi_{j,\cdot,h}}(\mathrm{x}_i, \mathrm{y}_i) + \left(1 - \frac{h}{h_{\max}}\right). \tag{7}$$

Our solution space $\mathcal{L}^0(\Theta, \mathcal{K}) = \bigcup_{h \in \mathcal{H}} \mathcal{L}_h^0(\Theta, \mathcal{K})$ for some Θ and $\mathcal{K}$ represents the collection of models under consideration. By the continuity of $\alpha(x, y)$, with $\mathcal{L}^{(\epsilon)} = \bigcup_{h \in \mathcal{H}} \mathcal{L}_h^{(\epsilon)}$, we know there exists some $\epsilon^* : \mathcal{L}^{(\epsilon^*)} \neq \emptyset$ and $\mathcal{L}^{(\epsilon)} = \emptyset\ \forall \epsilon < \epsilon^*$ (i.e., there exists an ϵ^* such that a solution exists when $\epsilon = \epsilon^*$, but no solution exists when $\epsilon < \epsilon^*$). To balance accuracy and robustness, we specify that at least 95 percent of the data points in D (not necessarily 100 percent) fall within ϵ^* of the optimal model—our solution belongs to $\mathcal{L}^{(\epsilon^*)}$. The choice of 95 percent is **heuristic**—it can be altered depending upon the characteristics of the dataset (e.g., the percentage of outliers) and the desired precision. Hence, our goal is to use genetic algorithms to find an optimal h^*-piecewise linear function $L(\boldsymbol{\theta}_{j^*,\cdot,h^*}, \boldsymbol{k}_{j^*,\cdot,h^*}) \in \{\mathcal{L}^0(\Theta, \mathcal{K}) \bigcap \mathcal{L}^{(\epsilon^*)}$ such that

$$\begin{aligned} ev_{\epsilon^*,N}\big(L(\boldsymbol{\theta}_{j^*,\cdot,h^*}, \boldsymbol{k}_{j^*,\cdot,h^*})\big) = & \\ \max_{L(\boldsymbol{\theta}_{j,\cdot,h}, \boldsymbol{k}_{j,\cdot,h}) \in \mathcal{L}^0(\Theta, \mathcal{K})} & ev_{\epsilon^*,N}\big(L(\boldsymbol{\theta}_{j,\cdot,h}, \boldsymbol{k}_{j,\cdot,h})\big), \end{aligned} \tag{8}$$

where $h^* = \min\{h \in \mathcal{H} : \mathcal{L}_h^{(\epsilon^*)} \neq \emptyset\}$.

We observe that if we require that the entire dataset D fall within ϵ^* of the optimal function, then, for Θ and $\mathcal{K}$ sufficiently large, the optimal solution will correspond to the least-squares solution.

Our goal can be attained if and only if

1. Our search space $\mathcal{L}^0(\Theta, \mathcal{K})$ contains an optimal solution as $\Theta \rightarrow \infty$ and $\mathcal{K} \rightarrow \infty$.

2. The algorithm converges to an optimal solution.

We will prove these statements by first assuming $h_{\max} = 1$ and then examining the case of $h_{\max} > 1$, h^* unknown, $h^* \in \mathcal{H}$.

4.4 Case $h_{\max} = 1$

4.4.1 Optimal Solution in Search Space

In the case where $h_{\max} = 1$ ($h^* = 1$), we would like our optimal string (solution) to represent a line $L(\theta_{j^*,\cdot,1}, k_{j^*,\cdot,1}) = L(\theta_{j^*,1,1}, k_{j^*,1,1})$ such that

$$ev_{\epsilon^*,N}\big(L(\theta_{j^*,1,1}, k_{j^*,1,1})\big) = \max_{L(\theta_{j,1,1},k_{j,1,1}) \in \mathcal{L}_1^0(\Theta,\mathcal{K})} ev_{\epsilon^*,N}\big(L(\theta_{j,1,1}, k_{j,1,1})\big).$$

Recall,

$$\mathcal{L}_1^0(\Theta,\mathcal{K}) = \Big\{ L(\theta_{j,1,1}, k_{j,1,1}) \in \mathcal{L}_1 : L(\theta_{j,1,1}, k_{j,1,1}) \text{ is of the form } x \cos\theta_{j,1,1} + y \sin\theta_{j,1,1} = \ell_{\theta_{j,1,1}} + k_{j,1,1}\delta, \text{where } \theta_{j,1,1} \text{ is one of } \Theta \text{ values, } k_{j,1,1} \text{ is one of } \mathcal{K} \text{ values} \Big\}.$$

Let

$$\mathcal{Q}_1 = \Big\{ L(\theta_{m,1,1}, k_{m,1,1}) : L(\theta_{m,1,1}, k_{m,1,1}) \in \mathcal{L}_1,\ \exists\ (x_r, y_r) \in rect \text{ satisfying } L(\theta_{m,1,1}, k_{m,1,1}),\ 0 < \theta_{m,1,1} \leq \pi,\ k_{m,1,1} \in \mathcal{R} \Big\}.$$

$\mathcal{Q}_1$ represents the set of all lines in $\mathcal{L}_1$ which intersect $rect$.

We will justify the following statements:

1. For any fixed $\epsilon \geq \epsilon^*$, our class $\mathcal{L}_1^0(\Theta,\mathcal{K})$ will contain an optimal line as $\Theta \to \infty$ and $\mathcal{K} \to \infty$.
2. For any fixed $\epsilon \geq \epsilon^*$, the set of optimal lines in $\mathcal{L}_1^0(\Theta,\mathcal{K})$ increases to the set of optimal lines from $\mathcal{Q}_1$ as $\Theta \to \infty$ and $\mathcal{K} \to \infty$.

   Additionally, we will show
3. For $0 < p \leq 1$, let

   $$\mathcal{A}^{(\epsilon)}(p) = \Big\{ L(\theta_{m,1,1}, k_{m,1,1}) \in \mathcal{Q}_1 : P\big(E_{\epsilon,L(\theta_{m,1,1},k_{m,1,1})}\big) \geq p \Big\}$$

   and let $\epsilon_p^* : \mathcal{A}^{(\epsilon_p^*)}(p) \neq \emptyset$ and $\mathcal{A}^{(\epsilon)}(p) = \emptyset\ \forall\ \epsilon < \epsilon_p^*$. Then for $p = 1.0$ and $N \to \infty$, $\mathcal{A}^{(\epsilon_p^*)}(p)$ converges to a unique optimal line $L^{1.0}(\theta_{j^*,1,1}, k_{j^*,1,1}) \in \mathcal{Q}_1$.

As the discretization of $\mathcal{L}_1$ becomes finer (i.e., as $\Theta \to \infty$ and $\mathcal{K} \to \infty$), Statement 1 says that our solution space will contain an optimal solution, while Statement 2 indicates that our solution space will contain all optimal solutions which intersect $rect$. Statement 3 says that as we require that a larger percentage of the data points fall within ϵ^* of the "fitted" function, the set of optimal functions intersecting $rect$ decreases to a single solution. This unique solution is the least-squares function.

Note that we have restricted ourselves to considering only those functions which intersect $rect$. This assumption is validated by the following theorem:

Theorem 4.1. *For any $\epsilon \geq \epsilon^*$and for Θ and $\mathcal{K}$ sufficiently large, there exists an optimal line $L(\theta_{j^{**},1,1}, k_{j^{**},1,1}) \in \mathcal{L}_1$ such that*

$$\big\{(x,y) : y = L(\theta_{j^{**},1,1}, k_{j^{**},1,1})(x), x \in [z_{11}, z_{21}]\big\} \bigcap\ rect \neq \emptyset.$$

Proof. A proof of Theorem 4.1. and a graphical representation of the proof are provided in the Appendix. □

From Theorem 4.1, we know that if an optimal function exists, then there exists an optimal function which intersects $rect$. It follows without loss of generality that, given $\epsilon \geq \epsilon^*$, we can restrict ourselves to Θ and $\mathcal{K}$ sufficiently large and only those optimal lines which intersect $rect$, i.e., those optimal lines which belong to $\mathcal{Q}_1$.

We now justify Statements 1, 2, and 3 with the help of the following propositions and theorems, the proofs of which can be found in the Appendix.

For simplicity, let $\rho = \ell_\theta + k\delta$ for given θ and k.

Statement 1 will be justified if we can show that, for Θ and $\mathcal{K}$ large, given any optimal line in $\mathcal{Q}_1$ we can find a line in $\mathcal{L}_1^0(\Theta,\mathcal{K})$ which is arbitrarily close to it. We begin with Proposition 4.1 which justifies that, given an optimal line in $\mathcal{Q}_1$, we can find a Θ and $\mathcal{K}$ such that there exists a line in $\mathcal{L}_1^0(\Theta,\mathcal{K})$ which is arbitrarily close to the given optimal line.

Proposition 4.1. *Let $L(\theta_{m,1,1}, k_{m,1,1}) \in \mathcal{Q}_1$. Let $\xi > 0$. Then, $\exists\ (\Theta_\xi, \mathcal{K}_\xi) : \forall\ \Theta > \Theta_\xi$ and $\mathcal{K} > \mathcal{K}_\xi$, $\exists\ L(\theta, k)$:*

1. *$L(\theta, k) \in \mathcal{L}_1^0(\Theta,\mathcal{K})$ and*
2. *$\mid \theta - \theta_{m,1,1} \mid < \xi/2$ and $\mid \rho_{m,1,1} - \rho \mid < \xi/2$.*

Now, given $\xi > 0$, we would like there to exist a Θ_ξ and $\mathcal{K}_\xi$ such that, given **any** optimal line in $\mathcal{Q}_1$ and **any** $\Theta > \Theta_\xi, \mathcal{K} > \mathcal{K}_\xi$, there exists a line in $\mathcal{L}_1^0(\Theta,\mathcal{K})$ which is arbitrarily close to the given optimal line. We know such a $(\Theta_\xi, \mathcal{K}_\xi)$ exists by the following theorem.

Theorem 4.2. *For each $\xi > 0$, $\exists\ (\Theta_\xi, \mathcal{K}_\xi)$: for all $\Theta > \Theta_\xi$ and for all $\mathcal{K} > \mathcal{K}_\xi$, given any $L(\theta_{m,1,1}, k_{m,1,1}) \in \mathcal{Q}_1$, $\exists\ L(\theta, k)$:*

1. *$L(\theta, k) \in \mathcal{L}_1^0(\Theta,\mathcal{K})$ and*
2. *$\mid \theta - \theta_{m,1,1} \mid < \xi/2$ and $\mid \rho_{m,1,1} - \rho \mid < \xi/2$.*

Thus, given Θ and $\mathcal{K}$ sufficiently large, we can get a line which is arbitrarily close to an optimal line. Hence, Statement 1 is justified.

For the purpose of justifying Statement 2, let $\{\Theta_i,\ i = 1, 2, \ldots\}$ and $\{\mathcal{K}_i,\ i = 1, 2, \ldots\}$ represent the possible values of Θ and $\mathcal{K}$. For any $\epsilon > 0$, we define

$$\mathcal{A}_{\epsilon i}(p) = \Big\{ L(\theta_{j_i,1,1}, k_{j_i,1,1}) \in \mathcal{L}_1^0(\Theta_i,\mathcal{K}_i) : P\big(E_{\epsilon,L(\theta_{j_i,1,1},k_{j_i,1,1})}\big) \geq p \Big\}$$

and

$$\mathcal{A}_\epsilon(p) = \Big\{ L(\theta_{m,1,1}, k_{m,1,1}) \in \mathcal{Q}_1 : P\big(E_{\epsilon,L(\theta_{m,1,1},k_{m,1,1})}\big) \geq p \Big\}$$

and let $\mathcal{A}_{\epsilon i} = \mathcal{A}_{\epsilon i}(0.95)$ and $\mathcal{A}_\epsilon = \mathcal{A}_\epsilon(0.95)$. $\mathcal{A}_{\epsilon i}$ represents the set of all optimal lines which belong to $\mathcal{L}_1^0(\Theta_i,\mathcal{K}_i)$, while $\mathcal{A}_\epsilon$ represents the set of all optimal lines which belong to $\mathcal{Q}_1$. We would like $\mathcal{A}_{\epsilon i} \to \mathcal{A}_\epsilon$ as $i \to \infty$. However, we first need that if $\Theta \to \infty$ and $\mathcal{K} \to \infty$, then, the limit of any sequence of functions in $\mathcal{Q}_1$ is contained in $\mathcal{Q}_1$ (and, hence, belong to our search space). This is, in fact, true, as stated in Proposition 4.2.

Proposition 4.2. *For each* $i = 1, 2, \ldots,$ *let*

$$L(\theta_{n_i,1,1}, k_{n_i,1,1}) \in \mathcal{L}_1^0(\Theta_i, \mathcal{K}_i) : \theta_{n_i,1,1} \rightarrow \theta_{\lim}$$

and $k_{n_i,1,1} \rightarrow k_{\lim}$ *for some* $\theta_{\lim}, 0 < \theta_{\lim} \leq \pi,$ *and* $k_{\lim}, 0 \leq k_{\lim} < \infty,$ *as* $i \rightarrow \infty$. *Let* $\mathcal{T}_1 = \{L(\theta_{\lim}, k_{\lim}) : \exists$ *a sequence*

$$\{L(\theta_{n_i,1,1}, k_{n_i,1,1})\}_{i=1}^{\infty}, L(\theta_{n_i,1,1}, k_{n_i,1,1}) \in \mathcal{L}_1^0(\Theta_i, \mathcal{K}_i)$$
$$\textit{such that } \theta_{n_i,1,1} \rightarrow \theta_{\lim} \text{ and } k_{n_i,1,1} \rightarrow k_{\lim}\}.$$

Then, the optimal lines in $\mathcal{Q}_1$ *are the optimal lines in* $\mathcal{T}_1$.

Hence, $\mathcal{Q}_1$ contains any optimal function which is a limit of a sequence of functions in $\mathcal{Q}_1$. We may now justify Statement 2.

Note that the fitness function $ev_{\epsilon,N} : (0, \pi] \times [-M, M] \rightarrow [0, \infty)$ is continuous where, for any line in $\mathcal{T}_1$, $\theta \in (0, \pi]$ and the distance of the line from the origin is less than M (i.e., $d \leq M$). With $ev_{\epsilon,N}$ continuous and bounded, we state Theorem 4.3.

Theorem 4.3. *Let* $\mathcal{A}_{\epsilon i}$, *and* $\mathcal{A}_\epsilon$ *be as defined above. Then,* $\mathcal{A}_{\epsilon i} \rightarrow \mathcal{A}_\epsilon$ *as* $i \rightarrow \infty$.

Hence, the optimal solutions which intersect *rect* are in the search space as $\Theta \rightarrow \infty$ and $\mathcal{K} \rightarrow \infty$.

The above theorem holds for any $\mathcal{A}_\epsilon(p)$, $0.95 \leq p \leq 1.0$, and for any $\epsilon \in \{\epsilon_p^*\}_{p=0.95}^1$, where $\epsilon_p^* : \mathcal{A}_{\epsilon_p^*}(p) \neq \emptyset$ and $\mathcal{A}_\epsilon(p) = \emptyset \ \forall \epsilon < \epsilon_p^*$ (i.e., if we require that 96 percent of the data points in a given dataset fall within ϵ of our optimal function, then $\epsilon_{0.96}^*$ is the smallest $\epsilon > 0$ for which such an optimal function exists). With this in mind, we conclude with a theorem which justifies Statement 3.

Theorem 4.4. *Let* $\mathcal{A}_\epsilon(p)$ ϵ_p^* *be as defined,* $0.95 \leq p \leq 1.0$. *Then,*

$$\mathcal{A}_{\epsilon_{1.0}^*}(1.0) \equiv L^{1.0}(\theta_{j^*,1,1}, k_{j^*,1,1}) \textit{ as } N \rightarrow \infty.$$

A graphical representation of this theorem is provided in the Appendix.

Suppose we require that all of the data points fall within ϵ of the "fitted" function, where ϵ is the smallest positive value for which such a function exists. Then, for Θ and $\mathcal{K}$ large, as $N \rightarrow \infty$ our optimal function will converge to the least-squares function. This implies that our algorithm can be used to fit nonrobust, as well as more robust optimal functions.

By justifying the above statements, we have shown that our search space contains an optimal solution. It remains to be shown that the algorithm converges to an optimal solution for $h_{\max} = 1$.

4.4.2 Convergence to Optimum

For any $\epsilon \geq \epsilon^*$, we know from the above section that we may choose Θ and $\mathcal{K}$ sufficently large so that an optimal solution is contained in the search space $\mathcal{L}_1^0(\Theta, \mathcal{K})$.

As defined previously, let

$$\mathcal{I}_{\epsilon,L(\theta_{j,1,1},k_{j,1,1})}(x, y) = \begin{cases} 1 & \mid y - L(\theta_{j,1,1}, k_{j,1,1}, x) \mid < \epsilon \\ 0 & \text{otherwise.} \end{cases}$$

Since h takes only one value, we may drop the term $(1 - \frac{h}{h_{\max}})$ from (7) and allow

$$ev_{\epsilon,N}\big(L(\theta_{j,1,1}, k_{j,1,1})\big) = \sum_{i=1}^{N} \mathcal{I}_{\epsilon,L(\theta_{j,1,1},k_{j,1,1})}(\mathrm{x}_i, \mathrm{y}_i).$$

Define $\overline{ev}_{\epsilon,N}\big(L(\theta_{j,1,1}, k_{j,1,1})\big) = \frac{1}{N} ev_{\epsilon,N}\big(L(\theta_{j,1,1}, k_{j,1,1})\big)$. Note that

$$\begin{aligned}\lim_{N\rightarrow\infty} \overline{ev}_{\epsilon,N}\big(L(\theta_{j,1,1}, k_{j,1,1})\big) &= P\Bigg(\bigcup_b \bigcup_{a<b} E_{\epsilon,L(\theta_{j,1,1},k_{j,1,1}),a,b}\Bigg) \\ &= P\Big(E_{\epsilon,L(\theta_{j,1,1},k_{j,1,1})}\Big).\end{aligned}$$

Given the convergence of the elitist GA, we know that our GA finds an optimal $L(\theta_{j,1,1}, k_{j,1,1})$ for a given $\mathcal{L}^0(\Theta, \mathcal{K})$, ϵ, and N. So, let $L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon,N}$ be such that

$$ev_{\epsilon,N}\Big(L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon,N}\Big) = \max_{L(\theta_{j,1,1},k_{j,1,1}) \in \mathcal{L}_1^0(\Theta,\mathcal{K})} ev_{\epsilon,N}\big(L(\theta_{j,1,1}, k_{j,1,1})\big),$$

where the dependence of an optimal line on ϵ and N has been made explicit. $L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon,N}$ also maximizes $\overline{ev}_{\epsilon,N}(L(\theta_{j,1,1}, k_{j,1,1}))$. To determine whether our algorithm converges to an optimal solution, we examine $\lim_{N\rightarrow\infty} \overline{ev}_{\epsilon,N}\Big(L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon,N}\Big)$ for $\epsilon = \epsilon_{2N}$, where

$$\overline{ev}_{\epsilon_{2N},N}(L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon_{2N},N}) \geq 0.95.$$

If $\lim_{N\rightarrow\infty} \overline{ev}_{\epsilon,N}(L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon,N})$ is at least 0.95, then our algorithm does indeed converge to an optimal solution. Theorem 4.5 states that this is indeed true.

Theorem 4.5. *Let* N *be large and* ϵ_{2N} *be as defined above. Then, for appropriate* Θ *and* $\mathcal{K}$,

$$\liminf_{N\rightarrow\infty} \overline{ev}_{\epsilon_{2N},N}\Big(L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon_{2N},N}\Big) \geq 0.95.$$

The proof of Theorem 4.5 is presented in the Appendix. Hence, we have established that, for $h_{\max} = 1$,

1. Our search space $\mathcal{L}^0(\Theta, \mathcal{K})$ contains an optimal solution as $\Theta \rightarrow \infty$ and $\mathcal{K} \rightarrow \infty$.
2. The algorithm converges to an optimal solution.

4.4.3 Remarks

1. For large Θ_{i_0} and $\mathcal{K}_{i_0}$, $\mid \theta_i - \theta_{i-1} \mid$ and $\mid k_i - k_{i-1} \mid$ are both small, so the optimal line $L(\theta_{j_{i_0}^*,1,1}, k_{j_{i_0}^*,1,1}) \in \mathcal{L}^0(\Theta_{i_0}, \mathcal{K}_{i_0})$ will be close to the optimal line $L(\theta_{m^*,1,1}, k_{m^*,1,1}) \in \mathcal{Q}$.
2. A search procedure based on the above mathematical formulation may be designed so that Θ, $\mathcal{K}$, and ϵ are adjusted adaptively, i.e., in a way that is dependent upon the algorithm's result. For example, we may start with initial choices Θ_{i_0}, $\mathcal{K}_{i_0}$, and ϵ_{i_0}, and run the algorithm for a finite number of iterations. If we reach several optimal results, we may reduce ϵ_{i_0}; if we reach a single result, we may increase Θ and $\mathcal{K}$. We then repeat this process until the resulting solution reflects the center of the probability density function of the observed random variables with an acceptable level of precision. Note that if Θ and/or $\mathcal{K}$ are/is small or ϵ is large, it is possible for the algorithm's result to be close to an optimum in terms

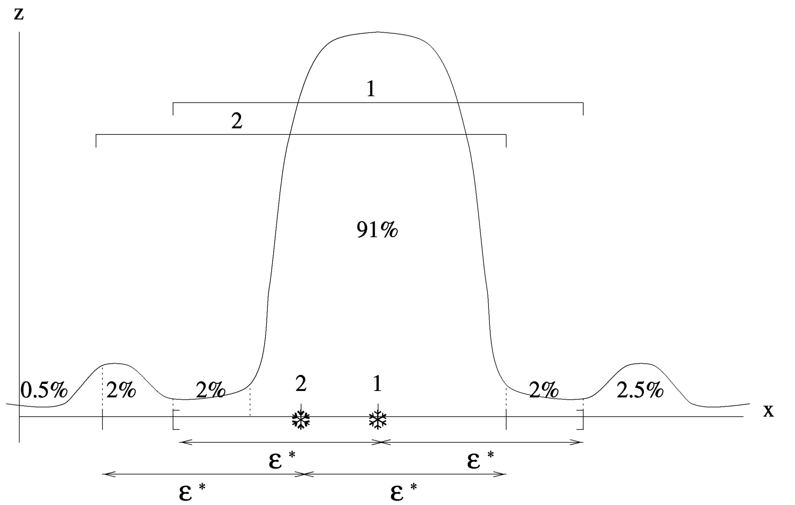


Fig. 2. The existence of two optimal lines for a given data set.

of probability but not in terms of Euclidean distance. Since appropriate values for Θ, $\mathcal{K}$, and ϵ are unknown a priori, we need to implement an **adaptive** procedure.

3. Recall that, in our requirement, 95 percent of the data points must be within ϵ^* of the optimal solution is **heuristic**. Depending upon the particular dataset at hand and the desired accuracy of the solution, we may alter this *critical level* to better suit the given situation.
4. In Statement 3 and its corresponding proof (Theorem 4.4), we justified that with a critical level of 1.0, the algorithm converges to a unique optimum as $N \rightarrow \infty$. Note that this unique optimum corresponds to the least-squares solution. Hence, our method can be used to fit both robust (using an ϵ criterion) and nonrobust (using a least-squares criterion) solutions.
5. It is possible for our optimization problem to have more than one solution. For example, consider Fig. 2, which shows a cross section of the density of the observed random variables (x, y). Suppose we have two parallel lines lying in the x-y plane, one located at the center of the interval marked with a number 1 (crossing the x-axis at the star marked with a number 1) and one located at the center of the interval marked with a number 2 (crossing the x-axis at the star marked with a number 2). The percent values on the graph indicate the percentage of the data points with an x value lying within the corresponding marked interval. Using these percentages, we see that 95 percent of the data points fall within ϵ of line number 1 and 95 percent of the data points fall within ϵ of line number 2. Hence, both of these lines would satisfy our optimization criteria (8).
6. We have assumed that the support of $\alpha_1(x, y)$, B, is rectangular in shape. However, the support of $\alpha_1(x, y)$ may have curved symmetric boundaries as opposed to straight lines. For example, let $\phi(x)$ be as defined in (6), i.e.,

$$\phi(x) = \begin{cases} \frac{d_1 - x\cos\theta_1}{\sin\theta_1} & x \in [z_{11}, z_{21}] \\ 0 & \text{otherwise.} \end{cases}$$

Then, we could have B as shown in Fig. 3. All of the above results except Theorem 4.4 hold for such a support B as long as the symmetricity of $\alpha_1(x, y)$ is maintained. The symmetricity is essential due to the nature of $ev_{\epsilon,N}$. For Theorem 4.4 to hold, we also require that $\gamma_1 \neq \gamma_2$ (so that the center portion of ϕ has positive width).

4.5 Case $h_{\max} > 1$

Having completed the case $h_{\max} = 1$, we now consider the case of $h_{\max} > 1$, h^* unknown, $h^* \in \mathcal{H}$. We extend the results of Section 4.4 to this case by utilizing the case of $h_{\max} > 1$, h^* known, $h^* \in \mathcal{H}$.

4.5.1 Optimal Solution in Search Space

Suppose $h = h^*$, $h^* > 1$, $h^* \in \mathcal{H}$, where h^* is known. Our optimal string should represent an h^*-piecewise linear function $L(\boldsymbol{\theta}_{j^*,\cdot,h^*}, \boldsymbol{k}_{j^*,\cdot,h^*})$ such that

$$ev_{\epsilon^*,N}\big(L(\boldsymbol{\theta}_{j^*,\cdot,h^*}, \boldsymbol{k}_{j^*,\cdot,h^*},)\big) = \max_{L(\boldsymbol{\theta}_{j,\cdot,h^*}, \boldsymbol{k}_{j,\cdot,h^*}) \in \mathcal{L}^0_{h^*}(\Theta,\mathcal{K})} ev_{\epsilon^*,N}\big(L(\boldsymbol{\theta}_{j,\cdot,h^*}, \boldsymbol{k}_{j,\cdot,h^*})\big),$$

where each $L(\theta_{j,i,h^*}, k_{j,i,h^*}), i = 1, \dots, h^*$, satisfies the properties listed in Section 4.2 for $L_{j,i,h}$, and

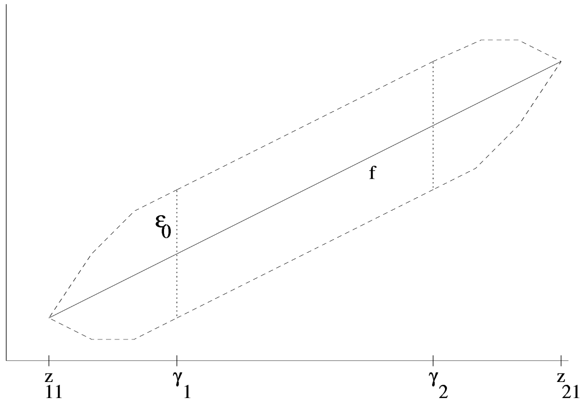


Fig. 3. A dataset whose density has a support B with curved boundaries.

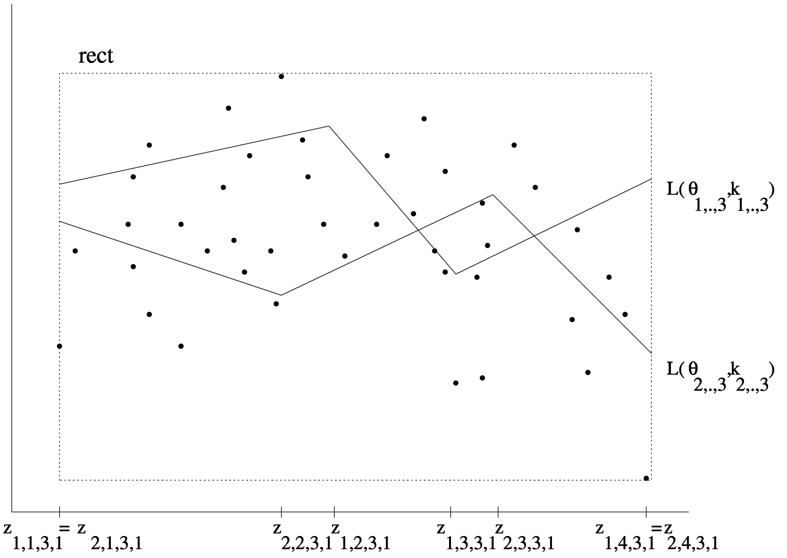


Fig. 4. Example functions from $\mathcal{L}_3^0$.

$$\mathcal{L}^0_{h^*}(\Theta,\mathcal{K}) = L_{j,\cdot,h^*} \ \in \mathcal{L}_{h^*} : L_{j,i,h^*} \text{ is of the form}$$

$$x\cos\theta_{j,i,h^*} + y\sin\theta_{j,i,h^*} = \ell_{\theta_{j,i,h^*}} + k_{j,i,h^*}\delta \ \ \forall\, \imath = 1,\dots,h^*,$$

$$\theta_{j,i,h^*} \in \frac{\pi}{2^{\mathrm{l_a}}}, \frac{2\pi}{2^{\mathrm{l_a}}}, \dots, \frac{(2^{\mathrm{l_a}}-1)\pi}{2^{\mathrm{l_a}}, \pi}, k_{j,i,h^*} \in \{0,1,\dots,2^{\mathrm{l_d}}-1\},$$
$$\text{and } \delta = diag/(2^{\mathrm{l_d}}-1)\}$$

(see Fig. 4).

With respect to optimization, we can approach each L_{j,i,h^*} as we did the $h_{\max} = 1$ case with $\alpha_i(x,y)$ as $\alpha_1(x,y)$ and $[z_{1i}, z_{1(i+1)}]$ as $[z_{11}, z_{21}]$. We then recognize $\mathcal{L}^0_{h^*}(\Theta,\mathcal{K})$ as simply $\bigcup_{i=1}^{h^*}\bigcup_j \mathcal{L}^0_{h^*,i,j}(\Theta,\mathcal{K})$, where

- $\mathcal{L}^0_{h^*,i,j}(\Theta,\mathcal{K})$ is $\mathcal{L}^0_1(\Theta,\mathcal{K})$ restricted to $[z_{j,1,h^*,1}, z_{j,2,h^*,1}]$ and
- the union is taken over all possible knot locations and over all possible pieces (any knot/piece combinations which contain pieces that do not intersect are removed).

By the results of Section 4.4.1, for any i, j we can get arbitrarily close to any line over $[z_{j,1,h^*,1}, z_{j,2,h^*,1}]$ intersecting *rect*, hence, we can get arbitrarily close to any piecewise function with knot locations satisfying $[z_{j,1,h^*,1},\dots,z_{j,(h^*+1),h^*,1}]$ whose pieces intersect *rect*. By taking the union over all possible knot locations satisfying $[z_{j,1,h^*,1},\dots,z_{j,(h^*+1),h^*,1}]$ and over all j, we see that an optimal h^*-piecewise solution is contained in $\mathcal{L}^0_{h^*}(\Theta,\mathcal{K})$ as $\Theta\to\infty$ and $\mathcal{K}\to\infty$.

By defining our search space as $\mathcal{L}^0(\Theta,\mathcal{K}) = \bigcup_{h\in\mathcal{H}}\mathcal{L}^0_h(\Theta,\mathcal{K})$ (see Section 4.3), we can guarantee, using the same reasoning as in Section 4.4.1, that an optimal h^*-piecewise solution, h^* unknown, $h^*\in\mathcal{H}$, will be contained in the search space as $\Theta\to\infty$ and $\mathcal{K}\to\infty$.

To show convergence to an optimal solution, we first consider the case of h^* known.

4.5.2 Convergence to Optimum

If h^* is known, $h^*\in\mathcal{H}$, then, for appropriate Θ and $\mathcal{K}$, we are searching for a function $L(\boldsymbol{\theta}_{j^*,\cdot,h^*}, \boldsymbol{k}_{j^*,\cdot,h^*}) \in \mathcal{L}^0_{h^*}(\Theta,\mathcal{K})\bigcap\mathcal{L}^{(\epsilon^*)}$ which maximizes

$$ev_{\epsilon,N}\big(L(\boldsymbol{\theta}_{j,\cdot,h^*}, \boldsymbol{k}_{j,\cdot,h^*})\big) = \sum_{i=1}^{N}\mathcal{I}_{\epsilon,L\left(\boldsymbol{\theta}_{j,\cdot,h^*},\boldsymbol{k}_{j,\cdot,h^*}\right)}(\mathrm{x}_i,\mathrm{y}_i) + \left(1-\frac{h^*}{h_{\max}}\right) \tag{9}$$

over all $L(\boldsymbol{\theta}_{j,\cdot,h^*}, \boldsymbol{k}_{j,\cdot,h^*}) \in \mathcal{L}^0_{h^*}(\Theta,\mathcal{K})$. As was done in Section 4.4.2, we may drop the term $(1-\frac{h^*}{h_{\max}})$. The convergence of the elitist GA combined with the continuity of $ev_{\epsilon,N}$ ensures that Theorem 4.5 holds, i.e.,

$$\lim_{N\to\infty}\inf \overline{ev}_{\epsilon_{2N},N}(L(\boldsymbol{\theta}_{j^*,\cdot,h^*}, \boldsymbol{k}_{j^*,\cdot,h^*})_{\epsilon_{2N},N}) \geq 0.95.$$

Since the above holds for any fixed $h = h^*$, an algorithm to find an optimal solution in the $h_{\max} > 1, h^*\in\mathcal{H}$ unknown case could be designed as follows:

1. Divide $\mathcal{L}^0(\Theta,\mathcal{K})$ into its component classes $\mathcal{L}^0_1(\Theta,\mathcal{K}), \mathcal{L}^0_2(\Theta,\mathcal{K}),\dots,\mathcal{L}^0_{h_{\max}}(\Theta,\mathcal{K})$.
2. On each class $\mathcal{L}^0_h(\Theta,\mathcal{K}), h = 1,\dots,h_{\max}$, use an elitist GA to find $L(\boldsymbol{\theta}_{j^*,\cdot,h}, \boldsymbol{k}_{j^*,\cdot,h})$, where
$$ev_{\epsilon,N}\big(L(\boldsymbol{\theta}_{j^*,\cdot,h}, \boldsymbol{k}_{j^*,\cdot,h})\big) = \max_{L(\boldsymbol{\theta}_{j,\cdot,h},\boldsymbol{k}_{j,\cdot,h})\in\mathcal{L}^0_h(\Theta,\mathcal{K})} ev_{\epsilon,N}\big(L(\boldsymbol{\theta}_{j,\cdot,h}, \boldsymbol{k}_{j,\cdot,h})\big).$$
3. Define
$$L_{best} = \big\{L(\boldsymbol{\theta}_{j^*,\cdot,1}, \boldsymbol{k}_{j^*,\cdot,1}),\dots,L(\boldsymbol{\theta}_{j^*,\cdot,h_{\max}}, \boldsymbol{k}_{j^*,\cdot,h_{\max}})\big\}.$$
Then, the solution is taken as the function $L(\boldsymbol{\theta}_{j^*,\cdot,h^*}, \boldsymbol{k}_{j^*,\cdot,h^*}) \in L_{best}$, where
$$ev_{\epsilon,N}\big(L(\boldsymbol{\theta}_{j^*,\cdot,h^*}, \boldsymbol{k}_{j^*,\cdot,h^*})\big) = \max_{L(\boldsymbol{\theta}_{j^*,\cdot,h},\boldsymbol{k}_{j^*,\cdot,h})\in L_{best}} ev_{\epsilon,N}\big(L(\boldsymbol{\theta}_{j^*,\cdot,h}, \boldsymbol{k}_{j^*,\cdot,h})\big).$$

This type of GA we call a *partitioned genetic algorithm*. We have noted previously that the elitist GA will converge to an optimal solution as $N \to \infty$. The proof of convergence is based on two assumptions:

1. The optimal string from the present population has a fitness value no less than the fitness values of the optimal strings from the previous populations.
2. Each string has a positive probability of going to an optimal string within any given iteration.

As these assumptions hold for the partitioned GA, it is easy to prove (although the proof will not be given here) that the proof of convergence to an optimal string holds for this algorithm as well.

4.5.3 Remarks

1. We have justified the claim that, for any value of $h_{\max}$, an optimal solution is contained in the search space of our algorithm and that our algorithm will converge to an optimal solution. We have also established that, for Θ and $\mathcal{K}$ large and $crit = 1.0$, the solution of our algorithm will converge to the least-squares solution as $N \to \infty$. Hence, our algorithm can be used to fit both robust and nonrobust solutions.
2. As in the $h_{\max} = 1$ case, for $h_{\max} > 1$, we do not know an appropriate value for ϵ a priori. Hence, to implement our algorithm, we must use an adaptive procedure which will search for an appropriate value for ϵ.

The theoretical foundation of our algorithm has been established. In the next section, we show the results of implementing our algorithm on several datasets and discuss how these results compare to those of similar methods.

5 Experimental Results

5.1 Methods and Implementation

5.1.1 Genetic Algorithm

We applied to each dataset a partitioned GA with either $\mathcal{H} = \{2, 3, 4\}$, $\mathcal{H} = \{3, 4, 5\}$, or $\mathcal{H} = \{5, 6, 7\}$. We utilized binary coding although an alternate coding, scheme could have been used. Since we chose values for Θ and $\mathcal{K}$ which were fixed, but very large (e.g., $\mathrm{l_a} = 8$ and $\mathrm{l_d} = 12$), the partitioned GA adaptively searched for only ϵ. We refer to such a GA as a *variable epsilon* genetic algorithm. For each value $h \in \mathcal{H}$, the variable epsilon GA can be described as follows:

1. Set the global parameters for population size M (≈ 50), crossover probability $p_c (p_c = 0.8)$, number of characters for representing angle $\mathrm{l_a}$ $(\mathrm{l_a} \approx 8)$, number of characters for representing distance or offset value $\mathrm{l_d}$ $(\mathrm{l_d} \approx 12)$, and the maximum number of iterations $MaxNit$ $(MaxNit \approx 20{,}000)$. Our selection for M depends on the computing power of our machine, while $\mathrm{l_a}$ and $\mathrm{l_d}$ depend upon the desired precision of our result. The mutation probability p_m was varied as a function of the number of iterations completed—see [30] for more details.
2. Choose $crit$ = critical level = percentage of data points to fall within ϵ of the final fitted piecewise linear function and a large initial value for ϵ. The fitness value of each string (which represents a function $L(\boldsymbol{\theta}_{j,\cdot,h}, \boldsymbol{k}_{j,\cdot,h}) \in \mathcal{L}^0(\Theta, \mathcal{K})$) is determined by (7).
3. For each $h \in \mathcal{H}$, run an elitist GA until either 1) $crit \leq$ (the maximum fitness value of the population)/(number of observations), or 2) the maximum number of iterations, $MaxNit$, is reached. If 1) occurs, then set $\epsilon = \epsilon - \tau(\tau \approx 0.01 * \epsilon), Nit =$ iteration number = 1, and restart the elitist GA. If 2) occurs, report the function corresponding to the string with the maximum fitness value as the final result for the given value of h.
4. Compare the results across h values and select the string corresponding to the overall maximum fitness value as the optimum string.

Note that, within each run of our algorithm, the value of h is fixed. We then compare the results of each run of the algorithm for each value of $h \in \mathcal{H}$ to determine the overall optimal string.

5.1.2 Data

The proposed algorithm was applied to a total of seven datasets; due to space constraints, we discuss results for those five datasets which best demonstrate the relative strengths and weaknesses of our method. These five datasets, of which three are simulated, are described in Table 1. Datasets 1 and 2 were generated from piecewise linear functions with unequally spaced knots using normally distributed noise (denoted Nor (mean, sd)); dataset 2 contains outliers while dataset 1 does not.

Dataset 3 was created using a generating function borrowed from Schwetlick and Schütze [14] with noise following a uniform distribution, as stated above.

The two nonsimulated datasets are standard datasets from the literature—the titanium heat data of DeBoor [31] and Pezzack et al.'s angular displacement data [16]. These were used to demonstrate the effectiveness of the proposed method in real applications.

5.1.3 Comparisons

For datasets 1 and 2, the results of the variable epsilon GA were compared to the results of two other piecewise linear fitting methods:

1. *Sb-spline*: A b-spline of degree 2 fit using the functions **bs** and **lm** in the software package S-plus, version 3.3, release 1 [32]. The knot locations are equally spaced by default.
2. *Least-squares GA:* A genetic algorithm which has the same global parameters as the variable epsilon GA, but which fits a piecewise linear function by minimizing the sum of squared errors.

All methods were applied to each dataset for each value $h \in \mathcal{H}$. Then, for each method, the results for all h values were compared and the best result was chosen as the method's overall solution.

Note that datasets 3, 4, and 5 have abscissae values in strictly increasing order. This allows comparison with more

TABLE 1
Experimental Datasets

Set #	N	h^*	Generating Function	Noise
1	320	3	$g(x) = \begin{cases} 3.88x + 10.44 & -3.0 \leq x < -1.29 \\ -1.74x + 3.14 & -1.29 \leq x < 3.7 \\ 3.77x - 17.25 & 3.7 \leq x \leq 4.9 \end{cases}$	$Nor(0, 1)$
2	50	4	$g(x) = \begin{cases} 1.3x + 2 & 0.0 \leq x < 0.9 \\ 0.05x + 3.125 & 0.9 \leq x < 2.1 \\ -0.4x + 3.2 & 2.1 \leq x < 4.3 \\ 3.2x - 11.41 & 4.3 \leq x \leq 5.0 \end{cases}$	$Nor(0, 0.2)$
3	128	—	$g(x) = \frac{10(4x-2)}{(1+(100*((4x-2)^2)))}$ $\quad 0 \leq x < 1$	$Unif(-0.1, 0.1)$
4	49	—	titanium hat data $\quad 595 \leq x \leq 1075$	———
5	142	—	Pezzack et al.'s data $\quad 0.0 \leq x \leq 2.834$	———

sophisticated spline fitting algorithms. Hence, for these examples, we will use the following comparative methods:

1. *deBoor/Rice*: A truncated power basis spline of degree 2 fit using the routines BSVLS, BSCPP, and BSLSQ in IMSL version 10 [33]. Given an initial set of knots, the knot locations are determined automatically by the nonlinear optimization algorithm of deBoor and Rice [15]. The number of knots is fixed by the user.
2. *dierckx*: A b-spline of degree 2 fit using the algorithm CURFIT of Dierckx [16]. The number of knots and their locations are chosen automatically by the algorithm; a single parameter must be specified to balance smoothness and lack of fit.

We have not attempted to compare results under neural network-based methods since we are fitting straight lines and not curves. A comparison of our results with those of a least-squares GA reflects the work of Karr and Weck [2] and Karr et al. [6]; a comparison with Vankeerberghen et al. least median squares GA [25] may be included in future research.

All experiments were run on a Sun Sparcstation 5.

5.2 Selected Results

5.2.1 Dataset 1

This dataset had a generating function with unequally spaced knots—the middle piece was considerably longer than either of the end pieces. For this reason, the fit of the Sb-spline was quite poor in comparison to the genetic algorithms' results. The best model from each method had the correct number of lines. The variable epsilon GA (with $crit$ = 95 percent) and the least-squares GA yielded similar results, followed by the Sb-spline. (see Fig. 5).

In some of our earlier experiments, the variable epsilon GA performed better than the least-squares GA. However, further experiments revealed that the least-squares GA does yield a comparable model given an equivalent number of iterations. The speed of the Sb-spline algorithm was almost instantaneous, while the GA algorithms took about five minutes of CPU time to yield a near optimal result (see Table 2).

With this dataset, all methods did merge lines for a better fit when the choice of h was too large. The Sb-spline could not match the quality of fit of the GAs because its knot locations were fixed.

5.2.2 Dataset 2

Dataset 2 combined unequally spaced knots with the presence of four outliers which were not clustered. We set $crit$ = 92 percent to accomodate for these outliers in our modeling procedure. From Fig. 6, it is clear that only the variable epsilon GA provided a reasonable fit. The least-squares GA was adversely affected by the outliers, while the Sb-spline failed to capture the shape of the dataset. The robustness of the variable epsilon GA proved essential for a proper fit. It should be noted, however, that the Sb-spline and the least-squares GA would in all likelihood fit better models in the presence of less influential outliers; see Section 5.4.

We also found that, when the number of lines was too large ($h = 4$ instead of $h = 3$), the variable epsilon GA was able to compensate for this by selecting pieces which almost merged.

5.2.3 Dataset 3

Dataset 3 has only one observation at each ordinate value and no outliers. Although our algorithm was designed with the robust case in mind, we feel that it is important for it to perform well in the nonrobust case. We set $crit$ = 99 percent and $\mathcal{H} = \{5, 6, 7\}$ (see Fig. 7).

It appears that the deBoor/Rice and variable epsilon GA algorithms, whose best models had $h = 5$ pieces, provided reasonable models, while the dierckx model was less adequate. We observed that, with this and the following datasets, the results of the deBoor/Rice algorithm were highly dependent upon the starting values; this is alluded to in [16]. However, given reasonable starting values, the algorithm yielded the models with the lowest error (see Table 3).

TABLE 2
Dataset 1, h = 3

Method	SSE	Method	SSE	Method	SSE
Sb-spline	610.24	lst. sq. GA	345.47	var. eps GA	351.57

We found the dierckx algorithm to be difficult to implement. It was found that, for an adequate fit, weights had to be supplied for the data points; balancing the weight values with the smoothing factor was found to be nontrivial. It was also noted that the number of knots in the dierckx model was considerably higher than the number of knots in the deBoor/Rice and variable epsilon models (in this case, we could not obtain a reasonable model for $h \in \mathcal{H}$; we required $h = 12$). It is possible, however, that an experienced user could find a more parsimonious model with an equal quality of fit.

5.2.4 Dataset 4

The titanium hat dataset is one of the standard datasets in the spline fitting literature. With $crit$ = 99 percent and $\mathcal{H} = \{5, 6, 7\}$, both the deBoor/Rice algorithm and variable epsilon GA selected $h = 5$; the dierckx method, in contrast, chose $h = 29$. In looking at Fig. 8 and Table 4, we note that the deBoor/Rice spline is the superior model, followed by the var. eps GA spline and the dierckx spline. Although the deBoor/Rice spline provides the best fit to the data, we note that the result of this algorithm was highly dependent upon the choice of initial knots. The result of the variable epsilon GA does not demonstrate this dependence.

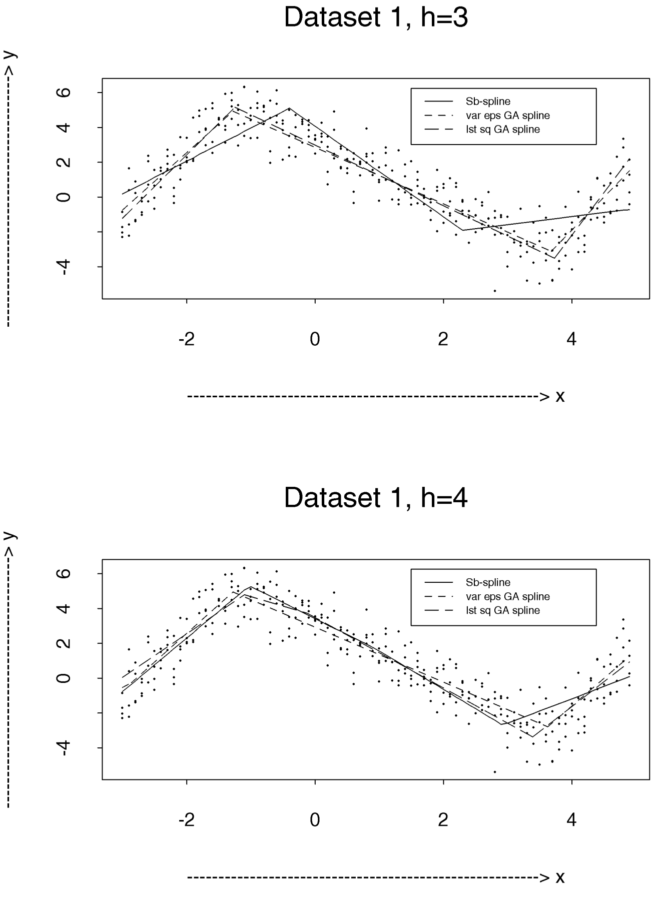


Fig. 5. Results of Dataset 1.

5.2.5 Dataset 5

Our last experimental dataset is Pezzack et al.'s angular displacement data [16], a popular dataset for testing curve fitting algorithms. Once again, the deBoor/Rice spline is the superior model, followed closely by the variable epsilon GA spline. Both models selected $h = 6$ from the candidate set $\mathcal{H} = \{5, 6, 7\}$. The dierckx model is relatively inadequate due to several spurious oscillations (this is reflected in the choice of $h = 74$). We attempted to remove these oscillations by increasing the smoothing factor, but this failed to improve the fit of the model. It is possible, however, that a more experienced user would have greater success in finding the correct balance between smoothing factor, data point weights, and lack of fit (see Table 5 and Fig. 9).

5.3 Conclusions

Our results demonstrate that, for "nice" datasets (no outliers), the variable epsilon GA can provide a fit comparable to that of a least-squares GA, deBoor/Rice, or dierckx spline model. Although the variable epsilon GA did not always yield the "best" result for nice datasets, it always achieved a satisfactory fit very quickly without the benefit of a user-defined initial knot set. When the number of pieces was too large, the algorithm did a nice job of decreasing the effective number of pieces by choosing lines which almost merged. By increasing the number of iterations and/or the string length, it is likely that even better results can be achieved.

For datasets with outliers, the variable epsilon GA appears capable of yielding results far superior to those of comparable methods. As with "nice" datasets, if too many pieces have been specified, the algorithm can almost merge pieces to yield a model with the appropriate number of effective knots. Despite large search spaces (of the order 2^{60} and 2^{80}), the algorithm can reach a near-optimal solution in about five minutes. The variable $crit$ makes it possible to adjust for outliers, while the variables Θ and $\mathcal{K}$ provide some control over the precision of the final fit. By allowing ϵ to be determined adaptively, our algorithm can find a result which 1) satisfies the critical value and 2) is closest to the majority of the data points with respect to other solutions.

We conclude that the variable epsilon GA represents a valuable tool for fitting both robust and nonrobust piecewise linear functions.

5.4 Remarks

1. It was noted in Section 5.2 that the results of methods based on least-squares, such as the algorithms of deBoor/Rice and dierckx, are adversely affected by the presence of outliers. We should note that this effect will be present most often when the outliers are *influential*. Hence, it is possible to use a least-squares method in the presence of such outliers

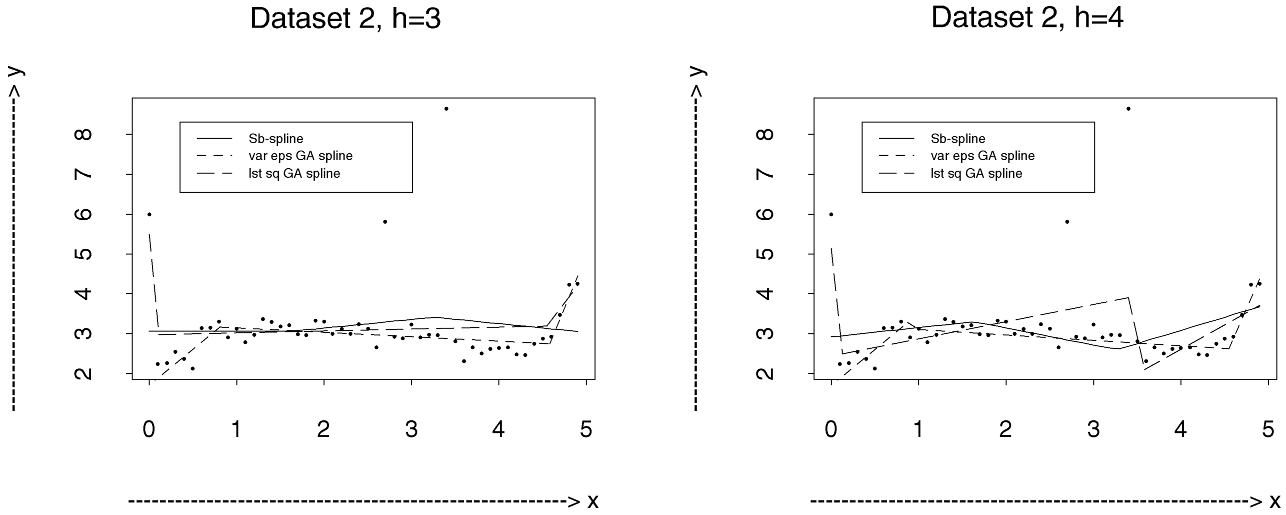


Fig. 6. Results of Dataset 2.

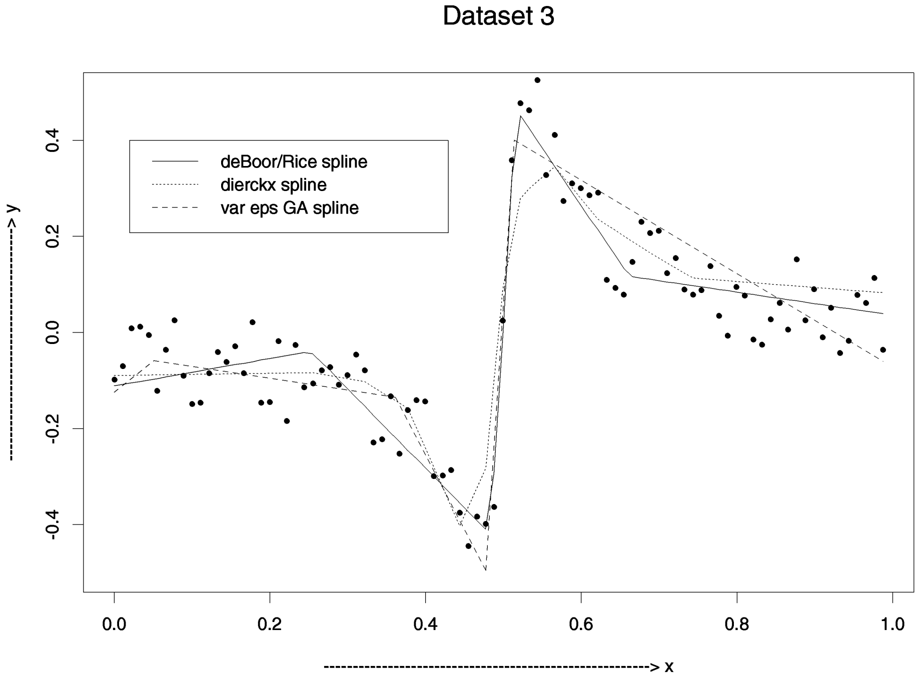


Fig. 7. Results of Dataset 3.

TABLE 3
Dataset 3

Method	SSE	Method	SSE	Method	SSE
deBoor/Rice	0.0034	dierckx	0.5886	var. eps GA	0.0041

if, e.g., one fits an initial model via least-squares, uses influence diagnostics [34] to test for influential observations, removes such observations from the analysis and then refits the model. This assumes, however, that such observations can be removed. It is often the case, especially in consulting situations, that the client refuses to remove any observations from the analysis. In such a situation, we would recommend that the analysis be performed using a method similar to the one presented here.

2. Since the solution given by the variable epsilon GA is the result of a random process, we decided to run the variable epsilon GA described above several times on the following dataset and examine the variability of the results. Dataset 6 contained several outliers, had equally spaced knots, and met the specifications in Table 6. A variable epsilon GA was executed eight times with the global parameters set, as in Section 5.1.1, and $crit = 95$ percent. The results are shown in Fig. 10 and Table 7.

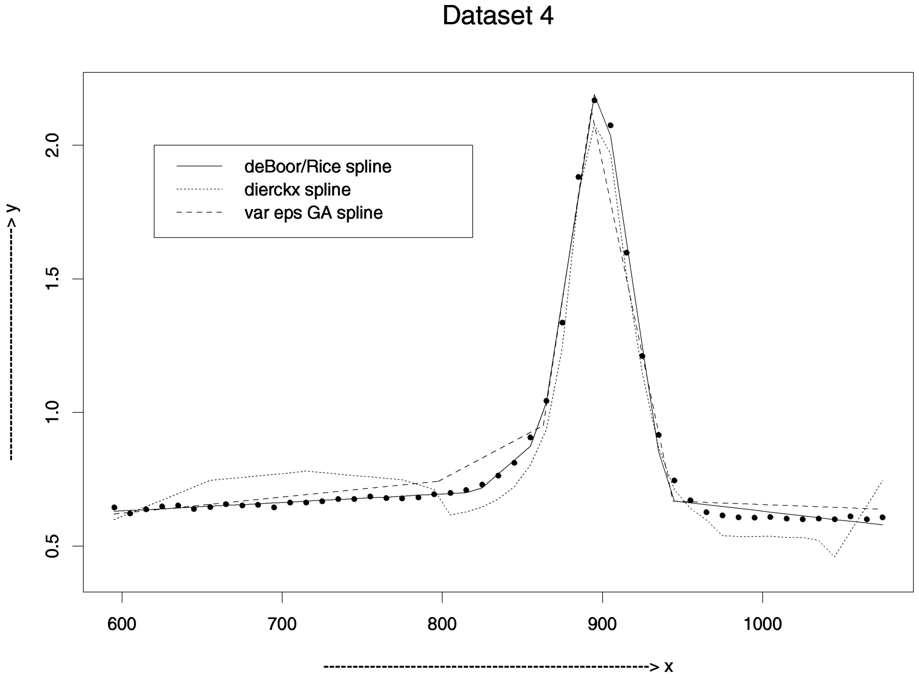


Fig. 8. Results of Dataset 4.

TABLE 4
Dataset 4

Method	SSE	Method	SSE	Method	SSE
deBoor/Rice	0.004	dierckx	0.3295	var. eps GA	0.044

Fig. 10 does not appear to contain eight functions because several results were identical. By examining the ranges of both the slopes and intercepts of the linear pieces, the achieved level of consistency appears to be satisfactory. Each parameter of the generating lines falls within its corresponding range in Table 7. For example, the slope and intercept of the first generating line are 2.5 and 0, respectively, and our experiments generated corresponding ranges of $(2.41421, 3.02704)$ and $(-0.188247, 0.180099)$. Further research is needed to determine a more precise measure of result variability.

3. For our experiments, we considered datasets where h was between 3 and 7. However, we have run experiments with larger values of h (≈ 10) and found that near optimal models can be found in a reasonable amount of time $(\approx 15$ min.). With the increasing availability of computational resources, it seems plausible that larger solution spaces (i.e., larger h and N) could be handled in approximately the same CPU time.

6 Conclusions and Future Research

By using genetic algorithms, we have devised a method for fitting piecewise linear functions to data in $\mathcal{R}^2$ which not only optimizes the number of pieces, but also optimizes the knot locations. With the assumption that the probability density function of our random variables is symmetric, the above theory shows that our method will lead to a piecewise linear function which "fits" the given dataset. However, even if we do not make this assumption (so our only assumption about the data is that the underlying probability density function is continuous), our method will still yield an optimal result for the chosen fitness function.

Our method yielded very good results in the presence of noise. The parameter $crit$ makes it possible for our algorithm to reach a near-optimal result even in the presence of outliers. The convergence of genetic algorithms has been shown for practically any choice of parameter values (e.g., $M = 50$, $p_c = 0.8$, etc.), although the best choices are still a matter of contention. The formulation of an optimal stopping rule is a subject of ongoing research, although it is known that increasing the number of iterations leads to a result with better accuracy.

It would be of interest to compare the performance of the partitioned GA with the performance of the VLGA. As mentioned previously, we would also like to develop a method for providing confidence limits for our results. It may also be possible to decrease CPU time by developing a heuristic for determining a "best" initial value for ϵ. Although our algorithm is currently restricted to linear splines in $\mathcal{R}^2$, future research will focus on extending our method to splines of higher orders, as in [36], and higher dimensions. Given the rapid increase in available computational power, we believe it is feasible to extend the use of genetic algorithms to fitting curvilinear models and to datasets of dimension greater than 2.

TABLE 5
Dataset 5

Method	SSE	Method	SSE	Method	SSE
deBoor/Rice	0.0147	dierckx	1.737	var. eps GA	0.0990

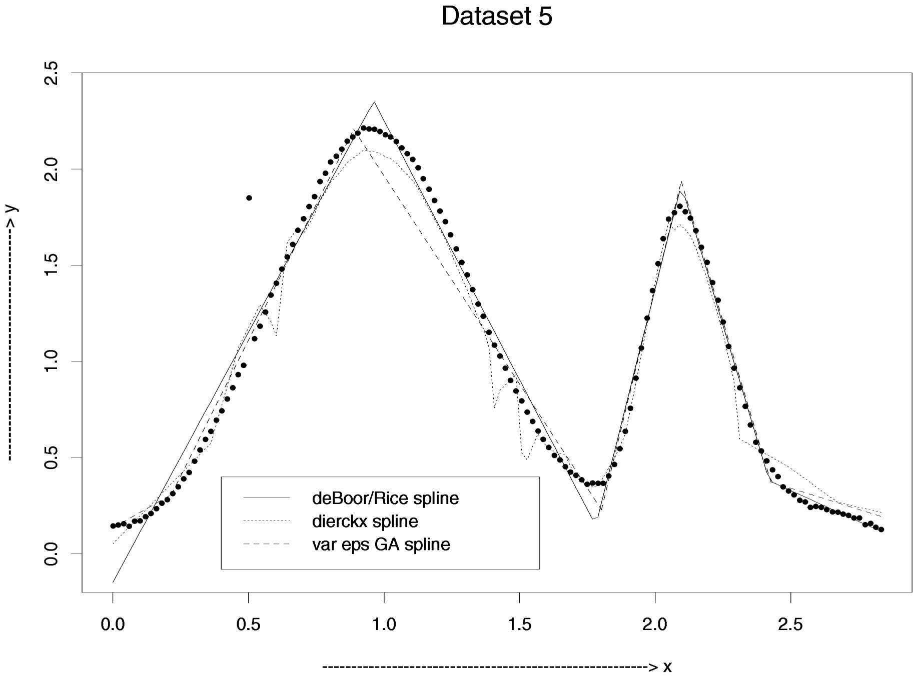


Fig. 9. Results of Dataset 5.

Appendix

Proof of Theorem 4.1. Let $\epsilon \geq \epsilon^*, \Theta,$ and $\mathcal{K}$ be given.

Suppose $\exists\ L(\theta_{j^*,1,1}, k_{j^*,1,1}) : L(\theta_{j^*,1,1}, k_{j^*,1,1}) \in \mathcal{L}^{(\epsilon)}$ and

$$\Big\{(x, y) : y = L(\theta_{j^*,1,1}, k_{j^*,1,1})(x),\ x \in [z_{11}, z_{21}]\Big\} \bigcap\ rect = \emptyset.$$

Then, either

$$\Big\{(x, y) : y \in \big[L(\theta_{j^*,1,1}, k_{j^*,1,1})(x), L(\theta_{j^*,1,1}, k_{j^*,1,1})(x) + \epsilon\big], x \in [z_{11}, z_{21}]\Big\} \bigcap\ rect = \emptyset$$

or

$$\Big\{(x, y) : y \in \big[L(\theta_{j^*,1,1}, k_{j^*,1,1})(x) - \epsilon, L(\theta_{j^*,1,1}, k_{j^*,1,1})(x)\big], x \in [z_{11}, z_{21}]\Big\} \bigcap\ rect = \emptyset.$$

Assume without loss of generality that

$$\Big\{(x, y) : y \in \big[L(\theta_{j^*,1,1}, k_{j^*,1,1})(x) - \epsilon, L(\theta_{j^*,1,1}, k_{j^*,1,1})(x)\big], x \in [z_{11}, z_{21}]\Big\} \bigcap\ rect = \emptyset.$$

This implies that $\frac{1}{N}\sum_{i=1}^{N} \varphi((\mathrm{x_i}, \mathrm{y_i})) \geq 0.95$, where

$$\varphi((x, y)) = \begin{cases} 1 & (x, y) \in \{(x, y) :\ y \in [L(\theta_{j^*,1,1}, k_{j^*,1,1})(x), \\ & \qquad L(\theta_{j^*,1,1}, k_{j^*,1,1})(x) + \epsilon], [z_{11}, z_{21}]\} \\ 0 & \text{otherwise.} \end{cases}$$

Let

$$(\bar{\bar{\mathrm{x}}}, \bar{\bar{\mathrm{y}}}) = (\sum_{i=1}^{N} \varphi((\mathrm{x}_i, \mathrm{y}_i)))^{-1} \sum_{j} (\mathrm{x}_j, \mathrm{y}_j),$$

where the sum is taken over all

$$(\mathrm{x}_j, \mathrm{y}_j) \in \Big\{(x, y) : y \in \big[L(\theta_{j^*,1,1}, k_{j^*,1,1})(x), L(\theta_{j^*,1,1}, k_{j^*,1,1})(x) + \epsilon\big], x \in [z_{11}, z_{21}]\Big\}.$$

Define

$$L(\theta_{j^{**},1,1}, k_{j^{**},1,1}) : L(\theta_{j^{**},1,1}, k_{j^{**},1,1})$$

is parallel to $L(\theta_{j^*,1,1}, k_{j^*,1,1})$ and $\bar{\bar{\mathrm{y}}} = L(\theta_{j^{**},1,1}, k_{j^{**},1,1})(\bar{\bar{\mathrm{x}}})$.

TABLE 6
Dataset 6

Set #	N	h	Function	Noise
5	60	3	$g(x) = \begin{cases} 2.5x & 0.0 \leq x < 1.0 \\ -4.0x + 6.5 & 1.0 \leq x < 2.0 \\ 3.5x + -8.5 & 2.0 \leq x \leq 3.0 \end{cases}$	$Nor(0, 0.1)$

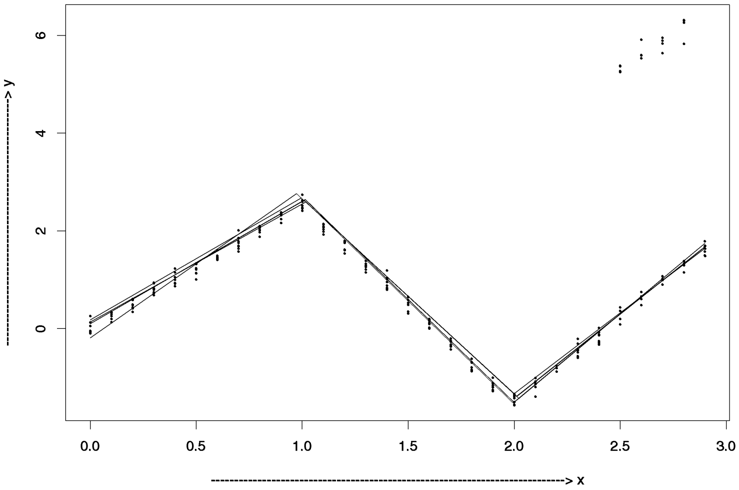


Fig. 10. Example of variability of results.

TABLE 7
Dataset 6 Results; Ranges for Intercepts and Slopes

	Piece #1	Piece #2	Piece #3
Intercept	(-0.188247, 0.180099)	(6.64789, 6.91933)	(-8.73460, -7.92068)
Slope	(2.41421, 3.02704)	(-4.21080, -3.99222)	(3.29656, 3.61354)

Then,

$$\Big\{(x,y) : y \in \big[L(\theta_{j^*,1,1}, k_{j^*,1,1})(x), L(\theta_{j^*,1,1}, k_{j^*,1,1})(x) + \epsilon\big], x \in [z_{11}, z_{21}]\Big\} \subseteq E_{\epsilon, L(\theta_{j^{**},1,1}, k_{j^{**},1,1})}.$$

Thus,

$$L(\theta_{j^{**},1,1}, k_{j^{**},1,1}) \in \mathcal{L}^{\epsilon}$$

and since

$$\bar{\bar{\mathrm{y}}} = L(\theta_{j^{**},1,1}, k_{j^{**},1,1})(\bar{\bar{\mathrm{x}}}),$$

$$\Big\{(x,y) : y = L(\theta_{j^{**},1,1}, k_{j^{**},1,1})(x),\ x \in [z_{11}, z_{21}]\Big\} \bigcap rect \neq \emptyset$$

□

For a graphical representation see Fig. 11a. Note that 95 percent of the datapoints fall within ϵ of $L(\theta_{j^*,1,1}, k_{j^*,1,1})$, while 95 percent of the datapoints fall within $\epsilon/2$ of $L(\theta_{j^{**},1,1}, k_{j^{**},1,1})$.

Proof of Proposition 4.1. Let $L(\theta_{m,1,1}, k_{m,1,1}) \in \mathcal{Q}_1$ and $\xi > 0$ be given. Choose $\Theta_\xi : \pi/2^{\mathrm{l_a}} = \pi/\Theta_\xi < \xi/2$. Similarly, choose

$$\mathcal{K}_\xi : \delta = diag/(2^{\mathrm{l_d}} - 1) = diag/(\mathcal{K}_\xi - 1) < \xi/2.$$

Then, $\exists\ L(\theta, k) \in \mathcal{L}_1^0(\Theta_\xi, \mathcal{K}_\xi)$: 2. is satisfied. Since, for any

$$h,\ h = 1, \ldots, h_{\max}, \{\mathcal{L}_h^0(\Theta, \mathcal{K}) : \mathrm{l_a} = 1, 2, \ldots\}$$

and

$$\{\mathcal{L}_h^0(\Theta, \mathcal{K}) : \mathrm{l_d} = 1, 2, \ldots\}$$

represent increasing sequences of nested sets, if $L(\theta, k) \in \mathcal{L}_1^0(\Theta_\xi, \mathcal{K}_\xi)$, then $L(\theta, k) \in \mathcal{L}_1^0(\Theta, \mathcal{K}) \forall\ \Theta > \Theta_\xi$ and $\forall\ \mathcal{K} > \mathcal{K}_\xi$. Hence, *1* is satisfied. □

Proof of Theorem 4.2. Let $\xi > 0$ be given. Choose $\Theta_\xi : \pi/2^{\mathrm{l_a}} = \pi/\Theta_\xi < \xi/2$. Then, for

$$\theta_{\xi_{(i)}} \in \left\{\frac{\pi}{2^{\Theta_\xi}, \ldots, \pi}\right\};\ \theta_{\xi_{(i)}} \leq \theta_{\xi_{(i+1)}}\ \forall\ i,\ i = 1, \ldots, \Theta_\xi - 1\},$$

we have $\mid 0 - \theta_{\xi_{(1)}} \mid < \xi/2$,

$$\mid \theta_{\xi_{(1)}} - \theta_{\xi_{(2)}} \mid < \xi/2, \ldots, \mid \theta_{\xi_{(\Theta_\xi)}} - \pi \mid < \xi/2.$$

So, given any $L(\theta_{m,1,1}, k_{m,1,1}) \in \mathcal{Q}_1$, we can choose Θ_ξ so that

$$\exists\ \theta_{\xi_{(i)}} \in \left\{\frac{\pi}{2^{\Theta_\xi}, \ldots, \pi}\right\} : \mid \theta_{m,1,1} - \theta_{\xi_{(i)}} \mid < \xi/2.$$

For any angle, $\theta_{\xi_{(n)}} \in \left[\frac{n\pi}{\Theta_\xi}, \frac{(n+1)\pi}{\Theta_\xi}\right], n = 1, \ldots, \Theta_\xi - 1$, the corresponding

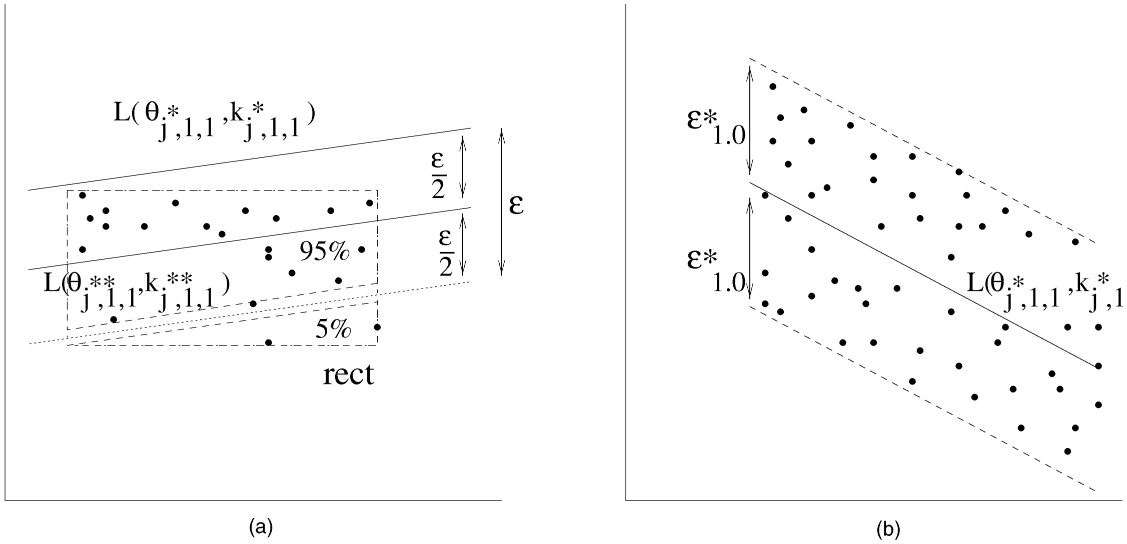


Fig. 11. Graphical representations of (a) Theorem 4.2 and (b) Theorem 4.4.

$$\rho_{\xi_{(n)}} \in [\gamma_{\xi_{(n_1)}}, \gamma_{\xi_{(n_2)}}], l_{\theta_{\xi_{(n)}}} \leq \gamma_{\xi_{(n_1)}} \leq \gamma_{\xi_{(n_2)}} \leq diag.$$

Find

$$\nu = \max_{n} \left\{ \sup_{\xi_{(n)}} \left[\gamma_{\xi_{(n_2)}} - \gamma_{\xi_{(n_1)}}\right],\ n = 1, \ldots, \Theta_\xi - 1 \right\}.$$

Choose $\mathcal{K}_\xi : \nu/\mathcal{K}_\xi < \xi/4$ and $\mathcal{K}_\xi = 2^y$ for some $y \in \mathcal{R}$. Then, given any $L(\theta_{m,1,1}, k_{m,1,1}) \in \mathcal{Q}_1$, we can choose $\mathcal{K}_\xi$ so that

$$\exists\, k_{\xi_{(i)}} \in \{0, \ldots, 2^{\mathcal{K}_\xi} - 1\} :\mid \rho_{\xi_{(i)}} - \rho_{m,1,1} \mid < \xi/2.$$

Hence, given any $\xi > 0$ and $L(\theta_{m,1,1}, k_{m,1,1}) \in \mathcal{Q}_1$, we can find Θ_ξ and $\mathcal{K}_\xi$ so that

$$\exists\, L(\theta_{\xi_{(i)}}, k_{\xi_{(i)}}) \in \mathcal{L}_1^0(\Theta_\xi, \mathcal{K}_\xi) : \\ \mid \theta - \theta_{m,1,1} \mid < \xi/2 \text{ and } \mid \rho_{m,1,1} - \rho \mid < \xi/2.$$

If $L(\theta_{\xi_{(i)}}, k_{\xi_{(i)}}) \in \mathcal{L}_1^0(\Theta_\xi, \mathcal{K}_\xi)$, then

$$L(\theta_{\xi_{(i)}}, k_{\xi_{(i)}}) \in \mathcal{L}_1^0(\Theta, \mathcal{K}) \forall\ \Theta > \Theta_\xi$$

and $\forall\ \mathcal{K} > \mathcal{K}_\xi$. Hence, *1* and *2* are satisfied. □

Proof of Proposition 4.2. Define $\mathcal{S}_1 = \bigcup_{i=1}^{\infty} \mathcal{L}_1^0(\Theta_i, \mathcal{K}_i)$. Note $\mathcal{Q}_1 \subseteq \mathcal{S}_1$. Note that $\mathcal{S}_1 \subseteq \mathcal{T}_1$ so $\mathcal{Q}_1 \subseteq \mathcal{T}_1$. Since we are only considering optimal lines which pass through *rect*, the optimal lines in $\mathcal{Q}_1 \equiv$ the optimal lines in $\mathcal{T}_1$. □

Proof of Theorem 4.3. Note that

$$\mathcal{L}_1^0(\Theta_i, \mathcal{K}_i) \subset \mathcal{L}_1^0(\Theta_{i'}, \mathcal{K}_{i'}) \,\forall 1' > i.$$

Hence, $\mathcal{A}_{\epsilon i} \subseteq \mathcal{A}_{\epsilon i'} \forall\ i' > i$. Thus, $\mathcal{A}_{\epsilon \lim} = \lim_{i \to \infty} \mathcal{A}_{\epsilon i}$ exists [35]. But, $\mathcal{A}_{\epsilon \lim}$ is the set of optimal lines in $\mathcal{S}_1$, where $\mathcal{Q}_1 \subseteq \mathcal{S}_1 \subseteq \mathcal{T}_1$. By Proposition 4.2, $\mathcal{A}_{\epsilon \lim} = \mathcal{A}_\epsilon$. □

Proof of Theorem 4.4. Suppose

$$\exists\, L(\theta_{j^*,1,1}, k_{j^*,1,1}), (\theta_{j^{**},1,1}, k_{j^{**},1,1}) \in \mathcal{A}_{\epsilon^*_{1.0}}(1.0) : \\ L(\theta_{j^*,1,1}, k_{j^*,1,1}) \neq L(\theta_{j^{**},1,1}, k_{j^{**},1,1}).$$

Then, for all possible values of (x, y)

$$(\text{x,y}) \in \Big\{(x, y) : \\ y \in \left[L(\theta_{j^*,1,1}, k_{j^*,1,1})(x) - \epsilon^*_{1.0},\ L(\theta_{j^*,1,1}, k_{j^*,1,1})(x) + \epsilon^*_{1.0}\right]\Big\}$$

and

$$(\text{x,y}) \in \Big\{(x, y) : \\ y \in \left[L(\theta_{j^{**},1,1}, k_{j^{**},1,1})(x) - \epsilon^*_{1.0},\ L(\theta_{j^{**},1,1}, k_{j^{**},1,1})(x) + \epsilon^*_{1.0}\right]\Big\}.$$

Recall that, for $h^* = 1$, the support of $\alpha_1(x, y)$ was defined as B, where

$$B = \left\{(x, y) : y \in \left[\frac{d_1 - x\cos\theta_1}{\sin\theta_1} - \epsilon_0, \frac{d_1 - x\cos\theta_1}{\sin\theta_1 + \epsilon_0}\right]\right\}$$

for $x \in [z_{11}, z_{21}]$.

As $N \to \infty$, $\epsilon^*_{1.0} \to \epsilon_0$ since $\epsilon^*_{1.0}$ is minimal and $ev_{\epsilon,N}$ is continuous for all ϵ. Hence,

$$\{(x, y) : y = L(\theta_{j^*,1,1}, k_{j^*,1,1})(x)\} \to \\ \{(x, y) : x\cos\theta_1 + y\sin\theta_1 = d_1\}$$

and

$$\{(x, y) : y = L(\theta_{j^{**},1,1}, k_{j^{**},1,1})(x)\} \to \\ \{(x, y) : x\cos\theta_1 + y\sin\theta_1 = d_1\}.$$

Thus, $L(\theta_{j^*,1,1}, k_{j^*,1,1}) = L(\theta_{j^{**},1,1}, k_{j^{**},1,1})$. □

See Fig. 11b for a graphical representation of Theorem 4.4.

Proof of Theorem 4.5. Given the continuity of $ev_{\epsilon,N}$, we know that, as $N \to \infty$, for each N large we may find $\epsilon_{1N}, \epsilon_{2N}, \epsilon_{3N}, 0 < \epsilon_{1N} < \epsilon_{2N} < \epsilon_{3N}$, such that:

- $\overline{ev}_{\epsilon_{1N},N}(L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon_{1N},N}) < 0.95$
- $\overline{ev}_{\epsilon_{2N},N}(L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon_{2N},N}) \geq 0.95$
- $\overline{ev}_{\epsilon_{3N},N}(L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon_{3N},N}) \geq 0.97$

Note that $L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon_{1N},N} \notin \mathcal{L}^{(\epsilon_{1N})}$, while ϵ_{3N} is not minimal, i.e., there exists some $\epsilon < \epsilon_{3N} : \mathcal{L}^{(\epsilon)} \neq \emptyset$. Hence, for our stated goal (see (8)), we are interested only in ϵ_{2N}. For each N, there may be infinitely many such ϵ_{2N}. For a given N and ϵ, let one such ϵ_{2N} be ϵ^*_{2N}. Then, we conclude,

$$\liminf_{N\to\infty} \overline{ev}_{\epsilon_{2N},N}(L(\theta_{j^*,1,1}, k_{j^*,1,1})_{\epsilon_{2N},N}) \geq 0.95$$

for appropriate Θ and $\mathcal{K}$. □

Acknowledgments

The research of J. Pittman is supported by the U.S. Army Research Office under Assert grant DAAG55-97-1-0219. The research of C.A. Murthy is supported by the U.S. Army Research Office under Assert grant DAAHO4-96-1-0082.

References

[1] P.A. Kokkonis and V. Leute, "Least Squares Splines Approximation Applied to Multicomponent Diffusion Data," *Computational Materials Science,* vol. 6, no. 1, pp. 103-111, 1996.

[2] C.L. Karr and B. Weck, "Improved Computer Modelling Using Least Median Squares Curve Fitting and Genetic Algorithms," *Fluid/Practice Separation J.,* vol. 8, no. 2, pp. 117-124, 1995.

[3] R.L. Eubank, *Spline Smoothing and Nonparametric Regression.* New York: Marcel Dekker, 1988.

[4] B. Cheng and D.M. Titterington, "Neural Networks: A Review from a Statistical Perspective," *Statistical Science,* vol. 9, pp. 2-30, 1994.

[5] K. Warwick and R. Craddock, "An Introduction to Radial Basis Functions for System Identification a Comparison with Other Neural Network Methods," *Proc. IEEE Conf. Decision and Control,* vol. 1, pp. 464-469 1996.

[6] C.L. Karr, D.A. Stanley, and B.J. Scheiner, "Genetic Algorithm Applied to Least-Squares Curve Fitting," Report of Investigations 9339, U.S. Dept. of the Interior, Bureau of Mines, 1991.

[7] S. Chatterjee, M. Laudato, and L.A. Lynch, "Genetic Algorithms and Their Statistical Applications: An Introduction," *Computational Statistics and Data Analysis,* vol. 22, no. 6, pp. 633-651, 1996.

[8] C.A. Murthy and J. Pittman, "Optimal Line Fitting Using Genetic Algorithms," Technical Report 99-02, Dept. of Statistics, The Pennsylvania State Univ., University Park, 1997.

[9] *Handbook of Genetic Algorithms,* L. Davis ed. New York: Van Nostrand Reinhold, 1991.

[10] D. Chen and R. Jain, "A Robust Back Propagation Learning Algorithm for Function Approximation," *IEEE Trans. Neural Networks,* vol. 5, pp. 467-479, 1994.

[11] B.D. Ripley, *Pattern Recognition and Neural Networks.* New York: Cambridge Univ. Press, 1996.

[12] R.L. Eubank and P. Speckman, "Curve Fitting by Polynomial-Trigonometric Regression," *Biometrika,* vol. 77, no. 1, pp. 1-9, 1990.

[13] P. Craven and G. Wahba, "Smoothing Noisy Data with Spline Functions," *Numerische Mathematik,* vol. 31, no. 4, pp. 377-403, 1979.

[14] H. Schwetlick and T. Schütze, "Least Squares Approximation by Splines with Free Knots," *BIT,* vol. 35, pp. 361-384, 1995.

[15] C. de Boor and J. Rice, "Least Squares Cubic Spline Approximation II—Variable Knots," CSD Technical Report 21, Dept. of Computer Science, Purdue Univ., West Lafayette, Ind., 1968.

[16] P. Dierckx, *Curve and Surface Fitting with Splines.* New York: Clarendon Press, 1993.

[17] L. Breiman, "Neural Networks: A Review from a Statistical Perspective," *Statistical Science,* vol. 9, p. 38, 1994.

[18] D.L. Barrow, C.K. Chui, P.W. Smith, and J.D. Ward, "Unicity of Best Mean Approximation by Second Order Splines with Variable Knots," *Math. Computing,* vol. 32, pp. 1,131-1,143, 1978.

[19] M.J. Baines, "Algorithms for Optimal Discontinuous Piecewise Linear and Constant $\mathcal{L}_2$ Fits to Continuous Functions with Adjustable Nodes in One and Two Dimensions," *Math. Computing,* vol. 62, no. 206, pp. 645-669, 1994.

[20] Y. Tourigny and M.J. Baines, "Analysis of an Algorithm for Generating Locally Optimal Meshes for $\mathcal{L}_2$ Approximation by Discontinuous Piecewise Polynomials," *Math. Computing,* vol. 66, no. 218, pp. 623-650, 1997.

[21] P.D. Loach and A.J. Wathen, "On the Best Least Squares Approximation of Continuous Functions Using Linear Splines with Free Knots," *IMA J. Numerical Analysis,* vol. 11, no. 3, pp. 393-409, 1991.

[22] C.K. Chui, P.W. Smith, and J.D. Ward, "On the Smoothness of Best $\mathcal{L}_2$ Approximants from Nonlinear Spline Manifolds," *Math. Computing,* vol. 31, pp. 17-23, 1977.

[23] D.G.T. Denison, B.K. Mallick, and A.F.M. Smith, "Automatic Bayesian Curve Fitting," *J. of the Royal Statistical Soc. Series B,* vol. 60, no. 2, pp. 333-350, 1998.

[24] D. Bhandari, C.A. Murthy, and S.K. Pal, "Genetic Algorithm with Elitist Model and Its Convergence," *Int'l J. Pattern Recognition and Artifical Intelligence,* vol. 10, no. 6, pp. 731-747, 1996.

[25] P. Vankeerberghen et al., "Robust Regression and Outlier Detection for Non-Linear Models Using Genetic Algorithms," *Chemometrics and Intelligent Laboratory Systems,* vol. 28, no. 1, pp. 73-87, 1995.

[26] S. Bandyopadhyay, C.A. Murthy, and S.K. Pal, "Pattern Classification Using Genetic Algorithms: Determination of $\mathcal{H}$," *Pattern Recognition Letters,* vol. 19, pp. 1,171-1,181, 1998.

[27] T. Bäck, "Optimal Mutation Rates in Genetic Search," *Proc Fifth Int'l Conf. Genetic Algorithms,* 1993.

[28] K.A. DeJong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," doctoral dissertation, Dept. of Electrical Eng. and Computer Science, Univ. of Michigan, Ann Arbor, 1975.

[29] P.J. Huber, *Robust Statistics.* New York: Wiley, 1981.

[30] S. Bandyopadhyay, C.A. Murthy, and S.K. Pal, "Pattern Classification with Genetic Algorithms," *Pattern Recognition Letters,* vol. 16, pp. 801-808, 1995.

[31] C. de Boor, *A Practical Guide to Splines.* New York: Springer-Verlag, 1978.

[32] R.A. Becker, "The New S Language," Bell Telephone Laboratories, Murray Hill, N.J., 1988.

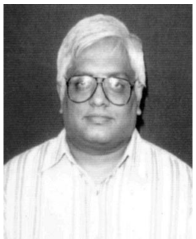
[33] Visual Numerics, Inc., *IMSL Fortran Numerical Libraries version 10.* 1997.

[34] S. Weisberg, *Applied Linear Regression.* New York: Wiley, 1985.

[35] R. Ash, *Real Analysis and Probability.* New York: Academic Press, 1972.

[36] J. Pittman, "Adaptive Splines and Genetic Algorithms," *J. of Computational and Graphical Statistics,* to appear.

Jennifer Pittman received Bachelors degrees in both mathematics and psychology from Skidmore College, Saratoga Springs, New York, in 1993, the Masters degree in statistics from Carnegie Mellon University, Pittsburgh, Pennsylvania, in 1995, and her PhD dissertation was written as an advisee of Professor C.R. Rao at The Pennsylvania State University and her degree conferred in 2000. Currently, she holds a postdoctoral position at the National Institute of Statistical Sciences (NISS), Research Triangle Park, North Carolina. Her research interests include evolutionary computation, adaptive splines, model selection, and capturing model uncertainty.



C.A. Murthy received his Masters and PhD degrees from the Indian Statistical Institute (ISI), Calcutta, India. During 1991 and 1992, he was a visting UNDP fellow at the Michigan State University, East Lansing. In 1996 and 1997, he was at the Center for Multivariate Analysis, Pennsylania State University. Currently, he is a professor in the Machine Intelligence Unit at ISI. His fields of interests include pattern recognition, image processing, computer vision, fuzzy logic, neural networks, fractals, genetic algorithms, and wavelets.