Triogram Models

Mark HANSEN, Charles KOOPERBERG, and Sylvain SARDY

In this article we introduce the Triogram method for function estimation using piecewise linear, bivariate splines based on an adaptively constructed triangulation. We illustrate the technique for bivariate regression and log-density estimation and indicate how our approach can be applied directly to model bivariate functions in the broader context of an extended linear model. The entire estimation procedure is invariant under affine transformations and is a natural approach for modeling data when the domain of the predictor variables is a polygonal region in the plane. Although our examples deal exclusively with estimating bivariate functions, the use of Triograms for modeling two-factor interactions in analysis of variance decompositions of functions depending on more than two variables is straightforward.

KEY WORDS: Adaptive triangulations; Density estimation; Extended linear models; Finite elements; Linear splines; Multivariate splines; Regression.

1. INTRODUCTION

Many of the spline-based solutions to multivariate estimation problems involve tensor product spaces that by necessity depend on the choice of coordinate system (see, e.g., Friedman 1991; Gu 1993; Kooperberg, Bose, and Stone 1997; Kooperberg, Stone, and Truong 1995; and Stone, Hansen, Kooperberg, and Truong 1997). Because these procedures are also highly adaptive, the estimates can change significantly when the data are rotated. This property is clearly undesirable when the given coordinate system is arbitrary, as is the case with spatial or compositional data. Moreover, the tensor product structure of these spaces implicitly defines the domain of an unknown function to be a hyperrectangle and can restrict the resulting estimators from capturing major features in the data that are not oriented along one of the major axes.

In the last 15 years, very few applications of multivariate splines other than tensor product spaces have appeared in statistical journals. During the same period, however, approximation theorists, numerical analysts and computer scientists have amassed a considerable body of literature on constructing and representing smooth, piecewise polynomial surfaces over meshes in many variables. In particular, much has been written about the case in which the underlying partition consists of triangles or high-dimensional simplicies. Because of their invariance to affine transformations, barycentric coordinate functions are often the starting point for constructing spline spaces over such meshes. In this article we develop a procedure for bivariate function estimation that borrows heavily from well-known properties of these coordinate functions.

Our estimates, christened *Triograms,* are continuous, piecewise linear functions defined over adaptively selected triangulations in the plane. The fitting is done via maximum likelihood, and the methodology can be applied to any of the so-called extended linear models, including density and conditional density estimation, generalized linear models, polychotomous regression, and hazard regression (Stone et al. 1997). In a process similar in spirit to stepwise knot addition and deletion in a univariate spline space, the underlying triangulation is constructed adaptively by adding and deleting vertices. These computations are made efficient through the use of the Rao (score) statistic for addition and the Wald statistic for deletion.

The spline spaces from which our Triogram models are built have existed in the approximation literature for many years. Data-driven adaptations to an underlying triangulation have been considered by many authors in the case of interpolation and least squares approximation. Our goal is to bring these ideas to the statistics community, and in so doing shift the focus to estimation problems. We revisit this issue in Section 3.5, where we make a more complete connection with approximation theory. Before describing our methodology in detail, we briefly discuss an example to illustrate the essential features of our Triogram models.

1.1 A Regression Surface With a Ridge

Cleveland and Fuentes (1996) analyzed data collected during an experiment to better understand the processing of liquid crystal mixtures. The response is the voltage V necessary to turn the material from opaque to clear. In our analysis we use two predictors: the percentage P of liquid crystal in the mixture and the temperature T of the mixture, measured in degrees Celsius. The experiment originally contained a third factor (the intensity of the light used in the processing) that was dropped half way into the experiment. Cleveland and Fuentes (1996) fitted a model consisting of two half planes that join along a line in T and P space. Triograms are a natural approach for automatically fitting such a piecewise planar model.

Figure 1a shows an initial triangulation together with the data points chosen by the experimenters. To this triangulation, we added vertices sequentially subject to the constraint that each triangle had to contain at least four data points. The largest model fit during this addition phase consisted

Mark Hansen is Member of the Technical Staff, Bell Laboratories, Murray Hill, NJ 07030. Charles Kooperberg is Associate Member, Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, Seattle, WA 98109. Sylvain Sardy is graduate student, Department of Statistics, University of Washington, Seattle, WA 98195. This work was done while Charles Kooperberg was at the Department of Statistics, University of Washington. Charles Kooperberg was supported in part by National Science Foundation grant DMS-9403371. The authors wish to thank William S. Cleveland and Charles J. Stone for many helpful discussions.

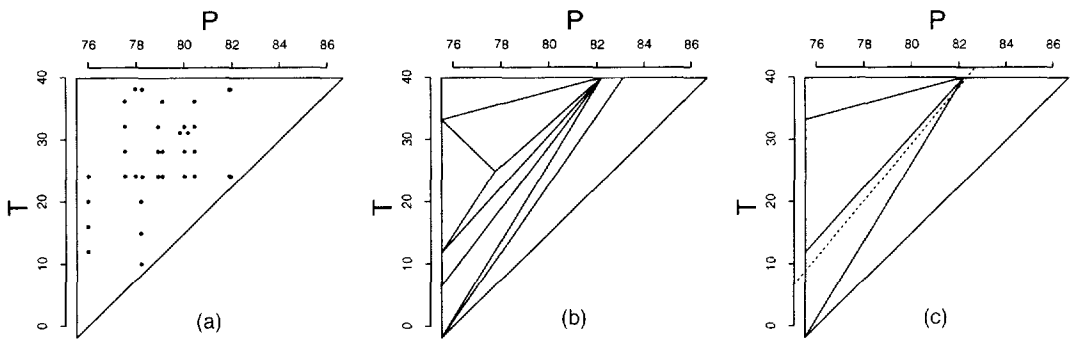


Figure 1. Initial Triangulation (a), Largest Triangulation (b), and Final Triangulation (c) for the Crystal Data. The dashed line in panel (c) is the edge fitted by Cleveland and Fuentes (1996).

of nine vertices, as shown in Figure 1b. From this maximal model, we deleted vertices sequentially, until we returned to the original triangulation. These addition and deletion steps generated a chain of nested models that we evaluated via generalized cross-validation (GCV). The best GCV model contained six vertices and is shown in Figure 1c. A perspective plot of this fit is given in Figure 2.

It is interesting to notice that the Triogram procedure puts an edge at almost the same location as the one fitted by Cleveland and Fuentes (the dashed line in Figure 1c). A second prominent feature of the Triogram surface is the downward fold near $T = 40$ and $P = 76$. Further data analysis suggested that just four observations were responsible for this fold. Through personal communication, the researchers who conducted this experiment indicated that these four observations were actually obtained from a different set of experiments than the remaining 43. In Section 4 we return to this example in more detail, comparing Triograms to other automatic procedures like multivariate adaptive regression splines (MARS).

Although this article focuses on estimating bivariate functions, Triograms can be applied much more generally. For example, let $\phi(u_1, u_2, u_3)$ denote an unknown function of three variables. Following the analysis of variance (ANOVA) framework developed by Hastie and Tibshirani (1990) and Stone (1994) for multivariate function estimation, subject to certain identifiability constraints, we can write

$$\phi(u_1, u_2, u_3) = \phi_0 + \phi_1(u_1) + \phi_2(u_2) + \phi_3(u_3) + \phi_{12}(u_1, u_2) + \phi_{13}(u_1, u_3) + \phi_{23}(u_2, u_3) + \phi_{123}(u_1, u_2, u_3). \quad (1)$$

By ignoring higher-order interaction terms in this expansion, the convergence rate of the remaining problem is governed by the largest of the dimensions of the terms that are estimated, taming the "curse of dimensionality." Polynomial splines can be used to model the main effect spaces, and tensor products of univariate spline spaces can be used to estimate the various interactions in the ANOVA decomposition (1). Alternatively, we can use the Triogram method presented in this article to model any of the two-dimensional components, or two-factor interactions. Although we ignore this possibility in most of this article, this is an important consideration in the theoretical framework for extended linear models developed by Hansen (1994), which we summarize in Section 5.

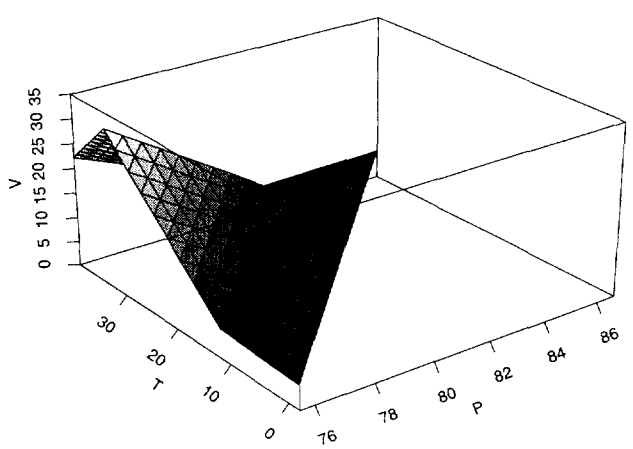


Figure 2. Triogram Fit for the Crystal Data.

In Section 2 we define barycentric coordinate functions and derive some basic facts to motivate their usefulness in representing polynomials over triangles in the plane. In Section 3 we introduce Triogram models together with general stepwise algorithms used to adaptively refine the underlying spline spaces. The barycentric coordinate functions again prove convenient for computing the various statistics required to perform this adaptation. In Section 4 we present a number of examples in which Triograms are used to construct bivariate regression and bivariate log-density estimates. As mentioned earlier, we reserve Section 5 for more technical remarks concerning theoretical rates of convergence for Triograms in the context of an extended linear model. Finally, in Section 6 we give some concluding remarks.

2. MULTIVARIATE SPLINES AND TRIANGULATIONS

We begin our discussion with a few definitions to cement our notation. Let $\mathcal{U}$ be a compact region in the plane, and let Δ be a collection of closed subsets of $\mathcal{U}$ having disjoint interiors satisfying

$$\mathcal{U} = \bigcup_{\delta \in \Delta} \delta.$$

The set Δ is said to form a tessellation of $\mathcal{U}$. If each element $\delta \in \Delta$ is a planar triangle, then Δ represents a triangula-

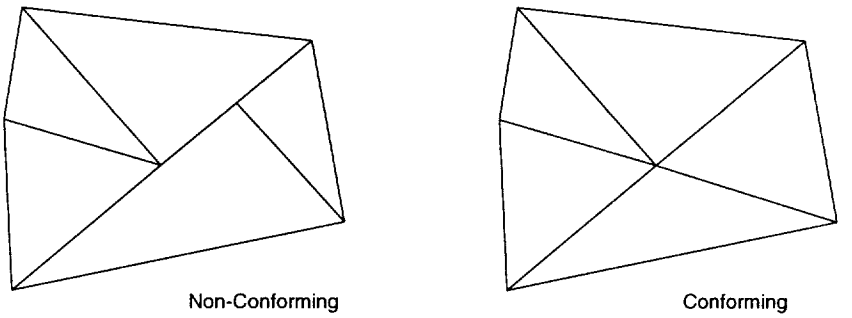


Figure 3. In a Nonconforming Triangulation, at Least One Vertex of a Triangle in Δ Falls Along the Interior of an Edge of Another Triangle in the Collection.

tion of $\mathcal{U}$. A triangulation Δ is said to be conforming if the nonempty intersection between pairs of triangles in Δ consists of either a single shared vertex or an entire common edge (see Fig. 3). Throughout this article, we reserve the symbol Δ for this special type of tessellation.

Let G denote the space of continuous, piecewise linear functions over a given triangulation Δ. Each $g \in G$ is continuous on $\mathcal{U}$, and the restriction of g to $\delta \in \Delta$ is a linear function. Defined in this way, G is a finite-dimensional linear space and there is a natural association between the vertices $\mathbf{v}_1, \ldots, \mathbf{v}_J$ of the triangles in Δ and a set of basis functions $B_1(\mathbf{u}), \ldots, B_J(\mathbf{u})$ that span G. Define $B_j(\mathbf{u})$ to be the unique function that is linear on each of the triangles in Δ and takes on the value 1 at $\mathbf{v}_j$ and 0 at the remaining vertices in the triangulation. This collection of tent functions was originally proposed by Courant (1943) and is frequently used in the finite element method. As we show at the end of this section, these simple elements have also been used as the starting point for defining multivariate splines of higher degrees (see Chui 1988; de Boor 1987; and Farin 1986).

Many of the important properties of this basis can be obtained from a local representation of the tent functions. For the moment, we focus our attention on a single triangle $\delta \in \Delta$ having vertices $\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$. The barycentric coordinates of any point $\mathbf{u} = (u_1, u_2) \in \mathbb{R}^2$ are defined as a triple $\boldsymbol{\varphi}(\mathbf{u}) = (\varphi_1(\mathbf{u}), \varphi_2(\mathbf{u}), \varphi_3(\mathbf{u}))$, such that

$$\mathbf{u} = \varphi_1(\mathbf{u})\mathbf{v}_1 + \varphi_2(\mathbf{u})\mathbf{v}_2 + \varphi_3(\mathbf{u})\mathbf{v}_3$$

and

$$\varphi_1(\mathbf{u}) + \varphi_2(\mathbf{u}) + \varphi_3(\mathbf{u}) = 1.$$

These conditions are equivalent to the following set of linear equations:

$$\begin{pmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \varphi_1(\mathbf{u}) \\ \varphi_2(\mathbf{u}) \\ \varphi_3(\mathbf{u}) \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ 1 \end{pmatrix}. \tag{2}$$

which can be solved explicitly using Cramer's method provided that δ has a nonempty interior. The solution to this system of equations is best expressed in terms of the function SignedArea$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, which we define by

$$\text{SignedArea}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) = \frac{1}{2} \begin{vmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ 1 & 1 & 1 \end{vmatrix}.$$

As its name suggests, the absolute value of SignedArea($\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$) is just the area of the triangle with vertices $\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$. Applying Cramer's method to the set of equations in (2), we find that $\varphi_1(\mathbf{u})$ is given by the ratio

$$\varphi_1(\mathbf{u}) = \varphi_1(u_1, u_2) = \frac{\text{SignedArea}(\mathbf{u}, \mathbf{v}_2, \mathbf{v}_3)}{\text{SignedArea}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)}. \tag{3}$$

This relationship is illustrated in Figure 4.

From the expression in (3), we see that the barycentric coordinates are linear functions of u_1 and u_2, where $\mathbf{u} = (u_1, u_2)$, and satisfy the interpolation conditions

$$\varphi_i(\mathbf{v}_j) = \begin{cases} 0 & i \neq j \\ 1 & i = j, \end{cases} \qquad i, j = 1, 2, 3; \tag{4}$$

hence the vertices $\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$ have barycentric coordinates (1, 0, 0), (0, 1, 0), and (0, 0, 1). Furthermore, from (3) we see that the points on the edge connecting $\mathbf{v}_2$ and $\mathbf{v}_3$ have barycentric coordinates of the form $(0, \alpha, 1-\alpha), \alpha \in [0, 1]$. In general, any point on the boundary of δ has at least one zero coordinate. The interpolation conditions (4) can be used to demonstrate that the functions $\varphi_1(\mathbf{u}), \varphi_2(\mathbf{u})$, and $\varphi_3(\mathbf{u})$ are linearly independent and hence constitute a basis of the space of linear functions of $\mathbf{u} = (u_1, u_2) \in \mathbb{R}^2$. Although it is customary in statistical applications to choose

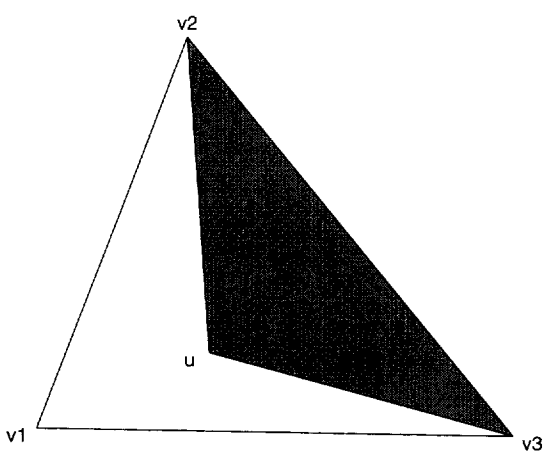


Figure 4. The Barycentric Coordinates of a Point **u** *Relative to the Triangle With Vertices* $\mathbf{v}_1$, $\mathbf{v}_2$, *and* $\mathbf{v}_3$ *are Expressed as Ratios of Signed Areas. In this case, the function* $\varphi_1(\mathbf{u})$ *is the ratio SignedArea(***u**, $\mathbf{v}_2$, $\mathbf{v}_3$*)/SignedArea(*$\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$*).*

the basis comprised of the constant function 1 and the two coordinate functions u_1 and u_2, the barycentric basis has the advantage that it is invariant under affine transformations such as rotations; given any nonsingular, 2×2 matrix $\mathbf{A}$ and any vector $\mathbf{b} \in \mathbb{R}^2$,

$$\varphi_i(\mathbf{u}) = \varphi_i^*(\mathbf{A}\mathbf{u} + \mathbf{b}), \quad \text{for} \quad i = 1, 2, 3 \quad \text{and} \quad \mathbf{u} \in \mathbb{R}^2. \quad (5)$$

where $\varphi_1^*(\mathbf{u}), \varphi_2^*(\mathbf{u})$, and $\varphi_3^*(\mathbf{u})$ are the barycentric coordinate functions of the vertices $\mathbf{A}\mathbf{v}_i + \mathbf{b}, i = 1, 2, 3$. For our applications, this means that the barycentric coordinate basis functions have a natural invariance under rotations and translations.

Returning to our triangulation Δ and the space of continuous, piecewise linear functions G, we let $\delta \in \Delta$ be a triangle with vertices $\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$ and observe that from the interpolation conditions (4), the functions $\varphi_1(\mathbf{u}), \varphi_2(\mathbf{u})$, and $\varphi_3(\mathbf{u})$ are exactly the basis functions $B_1(\mathbf{u}), B_2(\mathbf{u})$, and $B_3(\mathbf{u})$ for $\mathbf{u} \in \delta$. As an immediate consequence of this construction, we find that the basis of tent functions $B_1, \ldots, B_J$ associated with the triangulation Δ are bounded between 0 and 1 and satisfy

$$B_1(\mathbf{u}) + \cdots + B_J(\mathbf{u}) = 1, \qquad \mathbf{u} \in \mathcal{U},$$

a property shared by univariate B-spline bases. From (5), we also find that for any nonsingular, 2×2 matrix $\mathbf{A}$ and any vector $\mathbf{b} \in \mathbb{R}^2$,

$$B_j(\mathbf{u}) = B_j^*(\mathbf{A}\mathbf{u} + \mathbf{b}), \quad \forall\, \mathbf{u} \in \mathbb{R}^2,$$

where $B_1^*, \ldots, B_J^*$ is the basis associated with vertices $\mathbf{A}\mathbf{v}_1 + \mathbf{b}, \ldots, \mathbf{A}\mathbf{v}_J + \mathbf{b}$ of the transformed set $\mathcal{U}^* = \{\mathbf{A}\mathbf{u} + \mathbf{b}, \mathbf{u} \in \mathcal{U}\}$. This means that models built from functions in G have a natural invariance under affine transformations. Using the barycentric coordinate functions, we show in the next section that this invariance carries over to our adaptive methodology as well.

We have chosen to work with linear splines for a number of reasons. Even a brief survey of the literature on multivariate approximation theory indicates that there are many ways to generalize the classical univariate B-splines. Some procedures start with a triangulation Δ and attempt to construct smooth, piecewise polynomial basis functions that have small support by enforcing smoothness conditions across the edges in Δ. This finite element construction imposes rather severe restrictions on the resulting spline spaces even for functions in two variables. For example, given an arbitrary triangulation in the plane, any spline space consisting of functions with r continuous derivatives must have degree at least $3r + 2$ (see de Boor and Höllig 1988). This restriction can be alleviated somewhat by subdividing the triangles in Δ, but with added computational complexity (see Chui and He 1990). Other approaches define the mesh and the basis functions at the same time, a procedure analogous to "pulling apart knots" in a space of univariate B-splines. Recall that as knots coalesce in a univariate spline space, the functions have fewer continuous derivatives. One can envision reversing this process by starting with a space of discontinuous, piecewise polynomials having multiple knots at a single point and smoothing the space out by separating or pulling the knots apart. In the plane, one can start with discontinuous, piecewise polynomials over a triangulation Δ (see the description at the end of this section) that can be smoothed by separating multiple knots occurring at the vertices in the triangulation. Interestingly, in both the univariate and multivariate cases, the resulting functions can be described by considering marginal distributions of random vectors having support on high-dimensional polyhedra. The resulting polyhedral splines also come with considerable computational complexity (see Dahmen 1980 and de Boor 1976). (For a probabilistic interpretation, the reader is referred to Karlin, Micchelli, and Rinott 1986.) The simplest examples of this type of spline are the so-called box splines, which are defined with respect to very regular grids (see de Boor and Höllig 1982; de Boor, Höllig, and Riemenschneider 1993). We have chosen our space of tent functions because it is the starting point for these two approaches to spline construction. Additionally, data-driven rules for adaptively choosing Δ are more easily explored in this rather simple setting. We have more to say on this topic in Section 3.5, where we compare the Triogram algorithm with well-known techniques from approximation theory.

The price for this simplicity is that our Triogram estimates are crude. In Section 5 we make this notion precise by demonstrating how the L_2 rate of convergence for a nonadaptive version of our procedures depends on the approximation rate of the underlying spline space. By selecting linear splines, we are certain to suffer when estimating functions that are known to be very smooth. These suboptimal theoretical results for nonadaptive Triograms are less of a problem in practice, however, because our adaptive procedure uses the data to decide where to introduce new vertices. This effect was observed by Rippa (1992a) when he noted that even (theoretically) badly behaved triangulations consisting of long, thin triangles can have exceptional performance in bivariate interpolation problems when used in conjunction with an adaptive procedure.

As mentioned earlier, the barycentric coordinate functions can be used to generate spaces of higher-order polynomials defined relative to a triangle in the plane. For example, the space of quadratic polynomials spanned by the functions

$$1, u_1, u_1^2, u_2, u_2^2, u_1 u_2$$

is also spanned by the functions

$$\varphi_1^{i_1}(\mathbf{u})\varphi_2^{i_2}(\mathbf{u})\varphi_3^{i_3}(\mathbf{u}) \quad \text{for} \quad i_1 + i_2 + i_3 = 2, \quad (6)$$

where $\mathbf{u} = (u_1, u_2)$ and i_1, i_2, and i_3 are nonnegative integers. For polynomials defined over triangles, this basis is again more natural because of the invariance given in (5). When moving from a single triangle to a collection of triangles Δ, the B-net representation (Chui 1988; de Boor 1987; Farin 1986) can be used to define these basis functions so that the resulting spline spaces are continuous. Using this framework, elegant conditions can be derived to enforce higher-order smoothness across edges and vertices in Δ, reducing the task to a straightforward accounting problem (see Chui and Lai 1990). Although this procedure is still

subject to the severe conditions linking smoothness and degree, regular subdivision of Δ can also make use of the B-net structure to generate, for example, quadratic splines with continuous first partial derivatives in each coordinate direction over arbitrary triangulations (see Chui and He 1990).

3. TRIOGRAM MODELS

3.1 Maximum Likelihood Estimation

In the previous section, we derived some simple properties of a basis for the space G of continuous, piecewise linear functions defined over a conforming triangulation Δ of a region $\mathcal{U}$. In a Triogram model we estimate an unknown bivariate function $\phi^*(\mathbf{u}), \mathbf{u} \in \mathcal{U}$, as a member of G. To be more precise, let $\mathbf{W}_1, \ldots, \mathbf{W}_n$ be a random sample from the distribution of a random vector $\mathbf{W}$, and let $l(g, \mathbf{W}), g \in G$, denote the log-likelihood linking the distribution of $\mathbf{W}$ to functions in G. Using this notation, the Triogram estimate $\hat{\phi} \in G$ is given by

$$\hat{\phi} = \operatorname*{argmax}_{g \in G} l_n(g),$$

where

$$l_n(g) = \sum_{i=1}^{n} l(g, \mathbf{W}_i). \tag{7}$$

Equivalently, $l_n(g)$ can be written as $l_n(\boldsymbol{\beta})$, where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J) \in \mathbb{R}^J$ and

$$g(\mathbf{u}) = \beta_1 B_1(\mathbf{u}) + \cdots + \beta_J B_J(\mathbf{u}), \qquad \mathbf{u} \in \mathcal{U}.$$

Seen in this way, the estimate $\hat{\phi}$ is obtained by choosing the coefficients $\boldsymbol{\beta}$ that maximize the log-likelihood. In many cases, the random vector $\mathbf{W}$ can be partitioned into $(\mathbf{U}, \mathbf{V})$, where $\mathbf{U}$ is a random vector over $\mathcal{U} \in \mathbb{R}^2$ and $\mathbf{V}$ is a response vector.

We digress for a moment and present two simple examples to clarify these definitions.

Regression. Let $\mathbf{W} = (\mathbf{U}, V)$ with $V \in \mathbb{R}$ and set $\phi^*(\mathbf{U}) = E(V|\mathbf{U})$. Then, given observations $\mathbf{W}_1, \ldots, \mathbf{W}_n$, we estimate $\hat{\phi}(\cdot)$ by

$$\hat{\phi} = \operatorname*{argmax}_{g \in G} \sum_{i=1}^{n} (g(\mathbf{U}_i) - V_i)^2.$$

yielding the normal equations

$$\hat{\beta}_1 \langle B_i, B_1 \rangle_n + \cdots + \hat{\beta}_J \langle B_i, B_J \rangle_n = \langle B_i, V(\cdot) \rangle_n, \qquad 1 \le i \le J, \tag{8}$$

where $V(\mathbf{u}), \mathbf{u} \in \mathcal{U}$ is any function that interpolates the value V_i at $\mathbf{U}_i, 1 \le i \le n$; here for any two functions g_1 and g_2 defined on $\mathcal{U}$, we define the inner product $\langle \cdot, \cdot \rangle_n$ by

$$\langle g_1, g_2 \rangle_n = \frac{1}{n} \sum_{i=1}^{n} g_1(\mathbf{U}_i) g_2(\mathbf{U}_i).$$

By construction, the ith equation in (8) involves only those coefficients $\hat{\beta}_j$ for which the vertices $\mathbf{v}_i$ and $\mathbf{v}_j$ are joined by an edge in Δ. The maximum of $|i - j|$, taken over all pairs i, j such that $\mathbf{v}_i$ and $\mathbf{v}_j$ are connected by an edge in Δ, is referred to as the bandwidth of Δ. Schwarz (1988) described a number of well-known algorithms that renumber the vertices of an existing triangulation Δ to minimize its bandwidth. In our implementation of the Triogram fitting routine, we use one such procedure in conjunction with a band-limited Cholesky decomposition (Golub and Van Loan 1989) to solve the normal equations (8).

Density Estimation. Let ϕ^* represent the joint density of $\mathbf{U} \in \mathcal{U}$. In this context the vector $\mathbf{W}$ equals $\mathbf{U}$, because we do not have a response. Now, given coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J) \in \mathbb{R}^J$, we can define a density $f(\mathbf{u}; \boldsymbol{\beta})$ over $\mathcal{U}$ with the form

$$f(\mathbf{u}; \boldsymbol{\beta}) = \exp(\beta_1 B_1(\mathbf{u}) + \cdots + \beta_J B_J(\mathbf{u}) - C(\boldsymbol{\beta})),$$

where

$$C(\boldsymbol{\beta}) = \int_{\mathcal{U}} \exp(\beta_1 B_1(\mathbf{u}) + \cdots + \beta_J B_J(\mathbf{u}))\, d\mathbf{u}$$

is the normalizing constant. Therefore, based on a random sample $\mathbf{U}_1, \ldots, \mathbf{U}_n$ from the distribution of $\mathbf{U}$, we estimate ϕ^* by the function $\hat{\phi} = f(\cdot; \hat{\boldsymbol{\beta}})$, where $\hat{\boldsymbol{\beta}}$ is chosen to maximize the log-likelihood

$$l_n(\boldsymbol{\beta}) = \sum_{i=1}^{n} \log f(\mathbf{U}_i; \boldsymbol{\beta}).$$

As in univariate logspline density estimation (Kooperberg and Stone 1992), the likelihood equations take on the simple form

$$E_{\boldsymbol{\beta}} B_j(\mathbf{U}) = E_n B_j(\mathbf{U}), \qquad 1 \le j \le J, \tag{9}$$

where

$$E_{\boldsymbol{\beta}} B_j(\mathbf{U}) = \int_{\mathcal{U}} B_j(\mathbf{u}) f(\mathbf{u}; \boldsymbol{\beta})\, d\mathbf{u}$$

and

$$E_n B_j(\mathbf{U}) = \frac{1}{n} \sum_{i=1}^{n} B_j(\mathbf{U}_i).$$

Because the functions B_j are piecewise linear over $\mathcal{U}$, it is possible to evaluate the required integrals exactly, a definite advantage of Triogram models in the context of density estimation. In our Triogram software, Newton–Raphson iterations are used to solve the likelihood equations. To obtain the Hessian associated with this problem, we must compute quantities of the form $E_{\boldsymbol{\beta}}[B_{j_1}(\mathbf{U}) B_{j_2}(\mathbf{U})]$ for $1 \le j_1, j_2 \le J$, which again have closed-form expressions because our basis functions are piecewise linear.

The aforementioned examples serve both to cement notation and to highlight some computational advantages of Triogram modeling. Regression and density estimation are part of a larger class of extended linear models that also includes generalized regression, polychotomous regression,

and hazard regression (Stone et al. 1997). The methodology discussed in this article can be applied to any of the extended linear models when the unknown ϕ^* is a bivariate function defined on a domain $\mathcal{U}$. When ϕ^* depends on more than two variables, the Triogram methodology can be used to estimate two-factor interactions in a general ANOVA decomposition (Hansen 1994).

So far in this section, we have considered applying maximum likelihood to fit a Triogram model only for a fixed mesh Δ (and hence a fixed space G). In the remainder of this section, we describe a stepwise approach to Triogram model building that at each step alters an existing triangulation by adding or deleting a single vertex. After describing this algorithm in the context of estimation problems, we end this section by making connections between Triograms and similar adaptive procedures in the literature on approximation theory.

3.2 A Stepwise Algorithm

The adaptive Triogram procedure starts with an initial triangulation Δ_0 and a maximum likelihood estimate $\hat{\phi}_0 \in G_0$. In many applications a natural initial configuration may be determined by the shape of $\mathcal{U}$ or a priori knowledge about ϕ^*. For situations in which the initial triangulation is not so clearly defined, we provide several choices for Δ_0 in our Triogram software: the user can choose between the smallest triangle, the smallest equilateral triangle, and the smallest axis-oriented rectangle that contain all the data $\mathbf{U}_1, \ldots, \mathbf{U}_n$, with a possible magnification factor to avoid boundary problems. Note that only the procedures for determining the first two of these triangulations are invariant under affine transformations of the data. Figure 5 presents an example of each of these three initial triangulations corresponding to a random sample of 75 pairs of bivariate normal observations. From the discussion in the previous section, it is clear that for the first two configurations in this figure, the initial fit $\hat{\phi}_0$ is just a plane. In general, if the initial model is not sufficiently flexible to capture the major features of the data, then we enrich G_0 by stepwise refinements to the triangles $\delta \in \Delta_0$.

During the addition phase we produce a sequence of nested spaces $G_0 \subset G_1 \subset \cdots \subset G_m$ of continuous, piecewise linear functions with dimensions $p, p+1, \ldots, p+m$. As usual, associated with each space G_i is a conforming triangulation Δ_i of $\mathcal{U}$. Given the strong connection between vertices in a triangulation and the basis of tent functions described in the previous section, the most natural procedure for constructing the space G_{i+1} from G_i involves adding a single new vertex to the underlying triangulation Δ_i. There

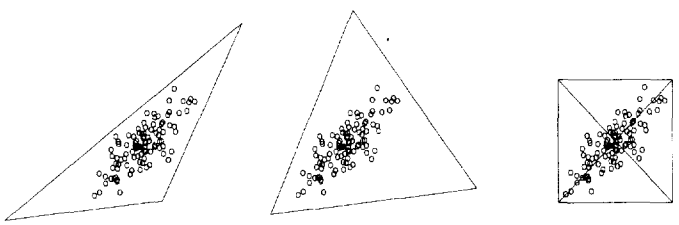


Figure 5. Three Standard Initial Triangulations.

are obvious constraints on this process, because the mesh Δ_{i+1} corresponding to G_{i+1} must also be a conforming triangulation, and G_i must be a subspace of G_{i+1}. In addition, we must only make changes to Δ_i that yield a space G_{i+1} in which the maximum likelihood equations (7) can be solved uniquely. For the moment, however, assume that at the ith stage in the addition process, we generate a number of candidate vertices that can be added to Δ_i to produce a refined triangulation Δ_{i+1} and a new space G_{i+1} representing a single degree-of-freedom change to G_i. We choose between these candidate vertices by a heuristic search that is designed approximately to maximize the Rao statistic (score statistic) associated with adding the corresponding new basis function. When ϕ^* is a regression function, for example, we select the vertex that produces the greatest decrease in the residual sum of squares when it is added to Δ_i. The user can specify the maximum number of vertices to add to an initial triangulation, and the addition phase continues until either this maximum is reached or we have exhausted the set of viable candidate vertices.

During the deletion phase of our Triogram procedure, we again construct a set of nested spaces $G'_0 \supset G'_1 \supset \cdots \supset G'_{m'}$, this time of decreasing dimension $p', p'-1, \ldots, p'-m'$. By again appealing to the close connection between vertices and basis elements in spaces of continuous piecewise linear functions, we see that the most natural process for generating these subspaces involves sequentially removing vertices from the maximal triangulation Δ'_0. This process is also subject to a number of constraints imposed by our requirements that G'_{i+1} be a subspace of G'_i and that the mesh associated with each space must be a conforming triangulation. Details about how vertices are identified as candidates for deletion are given in Section 3.4. For the purpose of this discussion, however, we simply assume that at each step i a number of vertices can be removed from Δ'_i to produce a smaller triangulation Δ'_{i+1} and a new space G'_{i+1} representing a single degree-of-freedom change to G'_i. From among these candidates, we choose the one that minimizes the Wald statistic associated with deleting the corresponding basis element from G'_{i+1}. For example, when ϕ^* is a regression function, we select the vertex that yields the least increase in the residual sum of squares when it is deleted from Δ'_i. As was the case with the addition phase, the user can specify the size of the smallest triangulation to be considered, and the deletion phase continues until either this minimum is reached or we have exhausted the set of viable candidate vertices.

By evaluating candidate vertices on the basis of Rao statistics during the addition phase and Wald statistics during the deletion phase, we avoid having to compute maximum likelihood estimates corresponding to each candidate space, improving the speed of our algorithm. Both statistics are based on quadratic approximations to the log-likelihood function (Stone et al. 1997). Regression is the only estimation context for which this does not represent a computational advantage, because the log-likelihood function is already quadratic.

During the combination of stepwise addition and stepwise deletion, we get a sequence of models indexed by ν,

with the νth model having p_ν parameters. When ϕ^* is a log-density function or a generalized regression function, the (generalized) Akaike information criterion (AIC) can be used to select the best model from this sequence. Let $\hat{l}_\nu$ denote the fitted log-likelihood for the νth model, and for a fixed penalty parameter a set

$$\mathrm{AIC}_{a,\nu} = -2\hat{l}_\nu + ap_\nu. \tag{10}$$

We take as our final model the member of the sequence that minimizes $\mathrm{AIC}_{a,\nu}$. In light of practical experience, we generally recommend choosing $a = \log n$ as in the Bayesian information criterion (BIC) due to Schwarz (1978), and set this as our default in the Triogram software. (Choosing $a = 2$ as in classical AIC tends to yield models that are unnecessarily complex, have spurious features, and do not predict well on test data.) When ϕ^* is a regression function, we discriminate between models on the basis of their GCV score (Friedman 1991)

$$\mathrm{GCV}_{a,\nu} = \frac{1}{n}\frac{\mathrm{RSS}_\nu}{\left(1 - \frac{ap_\nu}{n}\right)^2}. \tag{11}$$

where RSS_ν is the residual sum of squares for the νth model and a is a fixed penalty parameter. We select as our final model the member of the sequence that minimizes the GCV criterion. Note that we do not correct (11) for the number of parameters used in the initial model, because not all our initial models are of the same size. We have found that taking $a = 4$ approximately minimizes the mean squared error in a number of simulated examples, which agrees with the results of Friedman (1991), so this is our default choice in the Triogram software.

In the remainder of Section 3 we discuss in detail our implementation of the addition and deletion phases of an adaptive Triogram procedure, using many of the properties of the barycentric coordinate functions. Readers who are satisfied with the discussion given so far can safely skip to Section 4 for applications, or to Section 5 for an outline of the convergence properties of nonadaptive Triogram models.

3.3 Stepwise Addition

Inserting a new vertex into an existing triangulation Δ requires a rule for connecting this point to the vertices in Δ so that the new mesh is also a conforming triangulation. Figure 6 illustrates three options for vertex addition: We can place a new vertex on either a boundary or an interior edge, splitting the edge, or we can add a point to the interior of one of the triangles in Δ. Note that the space obtained by adding a vertex $\mathbf{v}$ to an interior edge of a triangle $\delta \in \Delta$ cannot be achieved as the limit of spaces constructed by adding $\mathbf{v}$ to the interior of δ. In this case, if $\mathbf{v}$ is very close to an edge of δ, then the new triangulation is essentially nonconforming, and the associated space of linear functions G contains elements that are discontinuous along that edge. Similar discontinuities arise when the new point $\mathbf{v}$ is positioned extremely close to an existing vertex. Degeneracies such as these are encountered in the context of univariate spline spaces when knots are allowed to coalesce (de Boor 1978).

Given a triangulation Δ, we construct a set of candidate vertices by considering the points with barycentric coordinates

$$\left(\frac{k_1}{K+1}, \frac{k_2}{K+1}, \frac{K+1-k_1-k_2}{K+1}\right)_\delta, \qquad \delta \in \Delta. \tag{12}$$

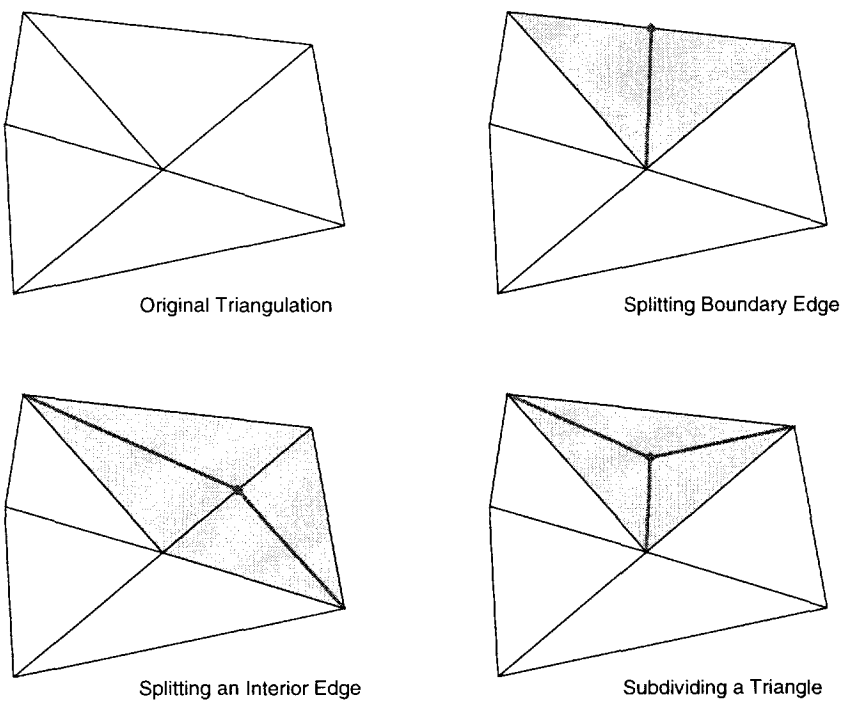


Figure 6. Three Ways to Add a New Vertex to an Existing Triangulation. Each addition represents the introduction of a single basis function, the support of which is colored gray.

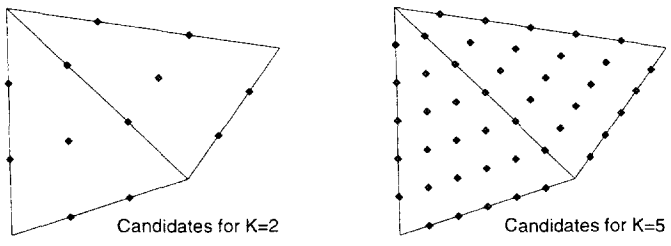


Figure 7. Candidate Vertices for K = 2 and K = 5.

where k_1, k_2, and K are nonnegative integers satisfying $k_1 + k_2 \leq K + 1$ and no coordinate equals 1. We have introduced a subscript "δ" to make it clear that these points are calculated for each triangle in Δ. Figure 7 plots the positions of the candidate knots calculated with $K = 2$ and $K = 5$ in (12). To avoid the aforementioned degeneracies, we suggest modest values of K, with 5 the default in our Triogram software. At this stage, we allow the user to impose other restrictions on the set of candidate vertices. For example, partitions Δ with many long, thin triangles or triangles containing little or no data tend produce highly unstable estimates. This notion is made precise in Section 5 when we examine the mean squared error properties of a nonadaptive Triogram procedure. For now, however, it is sufficient to indicate that the user can further restrict the set of candidate vertices by setting the minimum number of data points per triangle M and the minimum angle per triangle A in any allowable triangulation.

Recall that once we have identified a set of viable candidate vertices, we select the point that minimizes the Rao statistic. By evaluating a large number of potential vertices, we can generate a Rao surface that is useful in understanding both the behavior of the Triogram procedure as well as the placement of significant structures in a particular dataset. Figure 8 presents the Rao surface associated with adding a new vertex to a partition Δ consisting of just one triangle. In this case we are using ordinary least squares to estimate ϕ^*, the simple quadratic $u_1^2 + u_2^2$ plotted in Figure 8a. We generated 100 points uniformly in the triangle and added independent, normal noise to ϕ^* so that the signal-to-noise ratio was 3 to 1. Figure 8b presents the Rao surface for adding a new node to the triangle. Because we are estimating a regression function, the height of this surface at a particular point $\mathbf{u}$ is equivalent to the drop in the residual sum of squares when a new vertex is added to Δ at $\mathbf{u}$. Not surprisingly, it can be seen that the maximum Rao statistic is obtained when adding a vertex near the center of the triangle. In this example, the edges in the initial triangulation Δ form the boundary of U, and hence we do not observe any of the discontinuous features in the Rao surface associated with splitting interior edges.

Rather than choosing a new vertex from among a number of candidate vertices, we have also investigated the use of continuous, low-order polynomial approximations to the Rao surface. In this case, for each triangle $\delta \in \Delta$, we also calculate the Rao statistic at a small number of points following the recipe in (12), but fit a polynomial $\hat{p}_\delta(\mathbf{u})$ using the basis (6). The new vertex is then defined to be

$$\text{argmax}_{\mathbf{u} \in \delta} \hat{p}_\delta(\mathbf{u}) \quad \text{for} \quad \delta \in \Delta.$$

This approach allows for more flexibility in knot placement, with only minor computational overhead.

Once a new vertex has been identified, there is a simple procedure for generating the associated basis function $B(\mathbf{u})$, again using the barycentric coordinate functions described in Section 2. Suppose for the moment that we want to introduce a vertex $\mathbf{v}$ in the interior of a triangle δ with vertices $\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$. Recall that the barycentric coordinate functions $\boldsymbol{\varphi}(\mathbf{u}) = (\varphi_1(\mathbf{u}), \varphi_2(\mathbf{u}), \varphi_3(\mathbf{u})), \mathbf{u} \in \mathbb{R}^2$, associated with δ form a basis for the space of linear functions in $\mathbf{u} = (u_1, u_2)$. Therefore, any line in the plane can be expressed in the form

$$\alpha_1\varphi_1(\mathbf{u}) + \alpha_2\varphi_2(\mathbf{u}) + \alpha_3\varphi_3(\mathbf{u}) = 0, \qquad \mathbf{u} \in \mathbb{R}^2,$$

for suitable constants α_1, α_2, and α_3. In particular, the points $\mathbf{u}$ that lie on a line passing through the vertex $\mathbf{v}_1$ and any other point $\mathbf{v} \in \mathbb{R}^2$ are given by

$$\varphi_2(\mathbf{v})\varphi_3(\mathbf{u}) - \varphi_3(\mathbf{v})\varphi_2(\mathbf{u}) = 0, \qquad \mathbf{u} \in \mathbb{R}^2. \quad (13)$$

If $\mathbf{v}$ is contained in δ, then this line intersects the edge connecting $\mathbf{v}_2$ and $\mathbf{v}_3$, splitting δ into two subtriangles. The points $\mathbf{u} \in \delta$ satisfying $\varphi_2(\mathbf{v})\varphi_3(\mathbf{u}) \leq \varphi_3(\mathbf{v})\varphi_2(\mathbf{u})$ fall in the subtriangle that contains $\mathbf{v}_2$, whereas the remaining points in δ belong to the subtriangle containing $\mathbf{v}_3$. Similar statements can be made about lines connecting $\mathbf{v}$ and the other vertices $\mathbf{v}_2$ and $\mathbf{v}_2$.

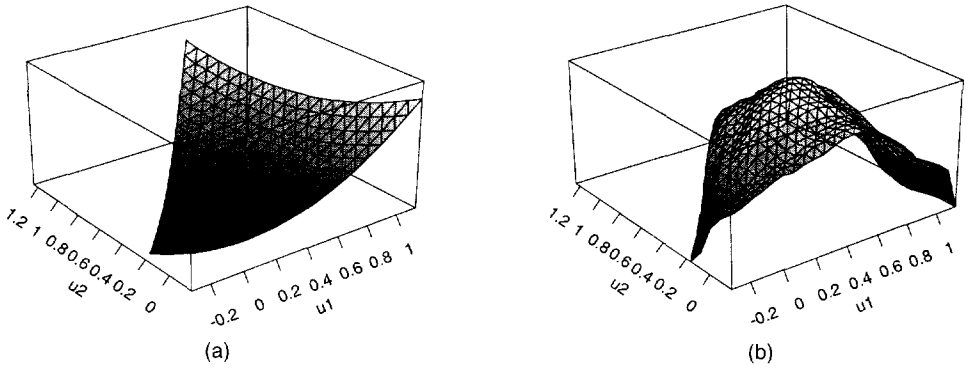


Figure 8. Rao Statistics for Adding Knot. (a) Truth; (b) the Rao surface for adding a single knot to simple linear fit.

With this relationship in mind, we define the quantities

$$\varphi_1^*(\mathbf{u}) = \frac{\varphi_1(\mathbf{u})}{\varphi_1(\mathbf{v})},$$

$$\varphi_2^*(\mathbf{u}) = \frac{\varphi_2(\mathbf{u})}{\varphi_2(\mathbf{v})},$$

and

$$\varphi_3^*(\mathbf{u}) = \frac{\varphi_3(\mathbf{u})}{\varphi_3(\mathbf{v})}.$$

From the discussion in the previous paragraph, we see that the points $\mathbf{u} \in \delta$ that fall within the triangular subregion with vertices $\mathbf{v}, \mathbf{v}_1$ and $\mathbf{v}_2$ (the shaded area in Fig. 9) satisfy the relationship $\varphi_3^*(\mathbf{u}) \le \varphi_1^*(\mathbf{u})$ and $\varphi_3^*(\mathbf{u}) \le \varphi_2^*(\mathbf{u})$. Applying (3) in Section 2, we also find that within this region, the new basis function $B(\mathbf{u})$ is given by $\varphi_3^*(\mathbf{u})$. Similar expressions can be derived for the remaining two subtriangles, yielding the following simple rule for constructing $B(\mathbf{u})$:

$$B(\mathbf{u}) = \begin{cases} \varphi_3^* & \text{if } \varphi_3^* \le \varphi_1^* \text{ and } \varphi_3^* < \varphi_2^* \\ \varphi_1^* & \text{if } \varphi_1^* \le \varphi_2^* \text{ and } \varphi_1^* < \varphi_3^* \\ \varphi_2^* & \text{if } \varphi_2^* \le \varphi_1^* \text{ and } \varphi_2^* < \varphi_3^*. \end{cases}$$

Using these expressions, it is easy to construct $B(\mathbf{u})$ from the existing basis elements associated with the vertices $\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$. When $\mathbf{v}$ is on the boundary of δ, at least one of the barycentric coordinates of $\mathbf{v}$ is 0. In this case, one of the φ_i^* must be infinite, and the foregoing conditions simplify. For example, if $\mathbf{v}$ is on the edge connecting $\mathbf{v}_1$ and $\mathbf{v}_2$, then φ_3^* is infinite, and we find that within δ,

$$B(\mathbf{u}) = \begin{cases} \varphi_1^* & \text{if } \varphi_1^* \le \varphi_2^* \\ \varphi_2^* & \text{if } \varphi_2^* < \varphi_1^*. \end{cases}$$

This set of equations creates $B(\mathbf{u})$ for $\mathbf{u} \in \delta$. If $\mathbf{v}$ is on the boundary of δ, then we might also have to produce a similar set of equations to construct $B(\mathbf{u})$ for $\mathbf{u}$ belonging to a neighboring triangle of δ. Because various inner products and empirical moments are already known for φ_1, φ_2, and φ_3 from the previous step in the addition process, these relationships can be used to derive simple updating formulas for computing the Rao statistic for adding $\mathbf{v}$ to the partition Δ.

Once a vertex has been chosen, we can again use the current barycentric coordinate functions to update the set of basis functions. Returning to the left hand triangle in Figure 9, suppose that we want to add a vertex $\mathbf{v}$ on the interior of δ. Now if we let $B_1(\mathbf{u}), B_2(\mathbf{u})$, and $B_3(\mathbf{u})$ represent the piecewise linear basis functions associated with the points $\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$ in the updated triangulation, then it is straightforward to demonstrate that for all points $\mathbf{u}$ in the shaded triangle on the right in Figure 9,

$$\varphi_1(\mathbf{u}) = B_1(\mathbf{u}) + \varphi_1(\mathbf{v})B_3(\mathbf{u}),$$

$$\varphi_2(\mathbf{u}) = B_2(\mathbf{u}) + \varphi_2(\mathbf{v})B_3(\mathbf{u}),$$

and

$$\varphi_3(\mathbf{u}) = \varphi_3(\mathbf{v})B_3(\mathbf{u}).$$

We have seen the last equation in the definition of the new basis function $B(\mathbf{u})$. Similar expressions can be obtained for the remaining two unshaded regions in δ and can be easily extended when $\mathbf{v}$ is on a boundary of δ. Again, because so much is known about φ_1, φ_2, and φ_3 from the previous step in the addition process, simple and efficient updating rules can be created for generating the new set of basis functions.

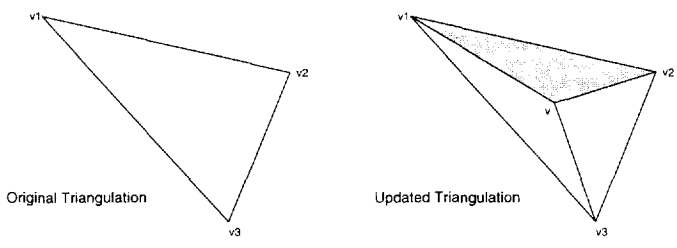


Figure 9. Adding a New Vertex at the Point $\mathbf{v} = \varphi_1(\mathbf{v})\mathbf{v}_1 + \varphi_2(\mathbf{v})\mathbf{v}_2 + \varphi_3(\mathbf{v})\mathbf{v}_3$. *In this case we are adding to G the continuous piecewise linear function that takes on the value one at the point* $\mathbf{v}$ *and 0 at each of* $\mathbf{v}_1$, $\mathbf{v}_2$, *and* $\mathbf{v}_3$.

3.4 Stepwise Deletion

When discussing strategies for reducing the dimension of a space of continuous, piecewise linear splines, so far we have only considered removing a vertex from an existing triangulation. In fact, this process can be viewed much more generally as enforcing continuity of the first partial derivatives along an edge in an existing triangulation. We now discuss both procedures in some detail.

Removing Vertices. In Figure 6 we outlined a rule that allows us to place a new vertex at any point in $\mathcal{U}$ to refine an existing triangulation. Unfortunately, when we remove a vertex from a partition Δ in an attempt to reduce the dimension of G, there may not be a way to reconnect the remaining vertices to form Δ_0 so that the updated space G_0 is a subspace of G. For example, the central vertex in any of the panels of Figure 6 cannot be removed if we want to obtain a subspace of G. Clearly, if any of the vertices highlighted in this figure are added to the initial triangulation in the upper left corner, then they can be immediately removed and still produce the proper nesting of spaces. Only vertices falling into one of the three categories listed in Figure 6 are legitimate candidates for removal in this restricted deletion strategy.

Enforcing Continuity of the First Partial Derivatives Along An Edge. This approach to stepwise deletion is more natural when we realize that removing a vertex amounts to enforcing the condition that a function in the space be continuously differentiable across a given edge in the existing triangulation. Observe that a continuous piecewise linear function has continuous partial derivatives across an edge if and only if the function is linear on the union of the two triangles that share the edge. In each of the examples in Figure 6, enforcing continuity of the first partial derivatives across any of the gray edges is equivalent to removing the added vertex, returning us to the original partition in the

upper left corner of the figure. These are the only cases for which this equivalence exists. (The strategy that we use in the examples in Section 4 involves using the Wald statistic to choose between continuity constraints across edges that fall into one of the three special categories.)

The alternative approach is somewhat more aggressive and involves choosing from among all the continuity constraints, regardless of how the edge is positioned relative to the other edges in the partition. The important distinction between these two procedures is that only in the first case are we actually guaranteed that the structure of Δ is simplified at each step.

Using the barycentric coordinate functions, we can derive a simple procedure for determining the constraint that a function in G be continuously differentiable across a given edge in Δ. To make this more precise, consider the triangulation on the left in Figure 10 and let $\varphi_1(\mathbf{u}), \varphi_2(\mathbf{u})$, and $\varphi_3(\mathbf{u})$ denote the barycentric coordinates of a point $\mathbf{u} \in \mathbb{R}^2$ relative to the triangle with vertices $\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$. Given a function $g \in G$, let β_1, β_2, and β_3 denote the coefficients of the basis functions associated with these vertices. Then for all points $\mathbf{u}$ in this triangle, $g(\mathbf{u})$ is the linear function given by $\beta_1\varphi_1(\mathbf{u}) + \beta_2\varphi_2(\mathbf{u}) + \beta_3\varphi_3(\mathbf{u})$. Now, if we let β_4 denote the coefficient of the basis function of G associated with the vertex $\mathbf{v}_4$, then $g(\mathbf{v}_4) = \beta_4$. Therefore, the function g is linear on the union of the two triangles in left hand portion of Figure 10 provided that

$$\beta_4 = g(\mathbf{v}_4) = \beta_1\varphi_1(\mathbf{v}_4) + \beta_2\varphi_2(\mathbf{v}_4) + \beta_3\varphi_3(\mathbf{v}_4).$$

By swapping the roles of $\mathbf{v}_1$ and $\mathbf{v}_4$ in this argument, we find that the C^1 continuity of a function $g \in G$ can also be assured by the constraint

$$\beta_1 = g(\mathbf{v}_1) = \beta_2\tilde{\varphi}_2(\mathbf{v}_1) + \beta_3\tilde{\varphi}_3(\mathbf{v}_1) + \beta_4\tilde{\varphi}_4(\mathbf{v}_1),$$

where $\tilde{\varphi}_2(\mathbf{u}), \tilde{\varphi}_3(\mathbf{u})$, and $\tilde{\varphi}_4(\mathbf{u})$ denote the barycentric coordinates of a point $\mathbf{u}$ relative to the triangle with vertices $\mathbf{v}_2, \mathbf{v}_3$, and $\mathbf{v}_4$. It is not hard to demonstrate that these two constraints are equivalent up to a multiplicative constant. Observe, however, that when this condition is enforced, we are left with a single linear function over the pair of triangles that constitute Δ, but we have not produced a simpler triangulation in the process.

Suppose instead that we want to remove the vertex $\mathbf{v}_4$ in the middle of the triangle in the right hand portion of Figure 10. Given $g \in G$ and $1 \le i \le 4$, we again let β_i correspond to the coefficient of the basis function associated with the vertex $\mathbf{v}_i$. It can be shown that each of the C^1 continuity constraints across the shaded interior edges shown in the figure is of the form

$$\beta_4 = \varphi_1(\mathbf{v}_4)\beta_1 + \varphi_2(\mathbf{v}_4)\beta_2 + \varphi_3(\mathbf{v}_4)\beta_3. \tag{14}$$

where $\varphi_1(\mathbf{u}), \varphi_2(\mathbf{u})$, and $\varphi_3(\mathbf{u})$ are the barycentric coordinates of a point $\mathbf{u}$ relative to the outer triangle on the right in Figure 10. Observe that the expression on the right is the value at $\mathbf{v}_4$ of the unique linear function interpolating β_1, β_2, and β_3 at the points $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$. Recalling that $g(\mathbf{v}_4) = \beta_4$, we see that the constraint in (14) has considerable intuitive appeal.

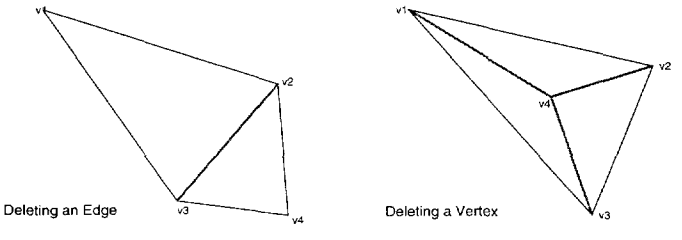


Figure 10. The Effect of Enforcing the Constraint That Functions in G be Continuously Differentiable Across Edges in Two Triangulations.

3.5 Triograms and Approximation Theory

As should be clear from the previous discussion, our choice of continuous piecewise linear splines has made the stepwise Triogram algorithm relatively easy to implement. It is precisely because of this computational simplicity that these spaces are at the heart of many well-known numerical procedures like finite element analysis. (In fact, procedures for *automatic* mesh or grid generation have a long history in the finite element literature, and the interested reader is referred to George 1991 for a discussion of the relevant issues.) Data-dependent adaptations to an underlying triangulation have also been suggested by a number of authors in the context of interpolation and approximation problems. For example, Dyn, Levin, and Rippa (1990a,b) and Rippa (1992b) obtained an "optimal" triangulation of a set of vertices $\mathbf{v}_1, \ldots, \mathbf{v}_J$ by successively swapping edges in some initial triangulation. In the configuration on the left in Figure 10, this operation involves removing the edge joining $\mathbf{v}_2$ and $\mathbf{v}_3$ and replacing it with a segment connecting $\mathbf{v}_1$ and $\mathbf{v}_4$. By creating various smoothness penalties for piecewise linear surfaces, Dyn et al. (1990a,b) swapped edges until they achieved a triangulation with minimum cost. Although originally designed for interpolation problems, this procedure has been applied to least squares approximation and has even been extended to include a vertex-addition phase in which data points (members of the collection $\mathbf{U}_1, \ldots, \mathbf{U}_n$ defined at the beginning of this section) are added until some specified approximation error has been achieved. Similar procedures have been considered by Quak and Schumaker (1991).

Closer in spirit to our Triogram algorithm, various knot addition and deletion procedures have been discussed by a number of authors. Again focusing on interpolation or least squares approximation, these procedures attempt to either simplify an existing piecewise linear surface or approximate a sampled bivariate function to within a given tolerance. Lyche (1993) gave an excellent review of the knot deletion schemes in this literature. The main strategy outlined in this survey involves ideas taken from Le Méhauté and Lafranche (1989, 1992) in which vertices are assigned weights that roughly correspond to the error incurred by removing the given vertex. Vertices are then removed sequentially, with low-valued points being removed first. To remain within the class of conforming triangulations, each removal necessitates a retriangulation, so that unlike our deletion procedure, it is not likely that triangulations ob-

tained from each deletion step nest. Similar ideas appear in the work of Hamann (1994), in which local curvature estimates are used to rank entire triangles for removal from an existing triangulation.

As for knot addition, Dierckx, Van Leemput, and Vermeire (1992) propose a technique based on a popular finite element refinement strategy (see Rivara 1984) that begins by splitting edges as in Figure 6. In this case, however, at each step they split not one but possibly many edges, introducing the extra structure to achieve a more "stable" triangulation. Ultimately, the authors want to ensure that in the final triangulation, the transitions between large and small triangles are gradual, and the smallest angle among the triangles is bounded from below. Hamann and Chen (1994) also have considered adding vertices, but (as in Hamann 1994) used local curvature measures to rank the candidate data points $\mathbf{U}_1, \ldots, \mathbf{U}_n$.

Although far from a complete survey of the literature, these references serve to highlight the fact that strategies for forming data-dependent triangulations have been considered in depth in the approximation literature. In addition, these citations serve to illustrate the differences between approximation and estimation. In the regression context, for example, we rarely have prior knowledge about the variance of the noise terms, and hence prespecified tolerances cannot be used for model selection. Furthermore, by considering nonlinear problems like density estimation, the models encountered at each stage must nest so that the computationally efficient Rao and Wald statistics can be used to decide between candidate vertices for addition or deletion. Our Triogram procedure borrows from the experience of numerical analysts, balancing, for example, the concept of a stable triangulation with the available degrees of freedom in the classic bias and variance trade-off. In the next section we apply the Triogram procedure to regression and density estimation problems to demonstrate the usefulness of these techniques in statistical applications.

4. EXAMPLES

In this section we present examples to illustrate the Triogram methodology. Our first three applications are each regression problems. We begin by studying how our procedure performs on data simulated from a model that has been widely studied in the literature on surface estimation. The next dataset was obtained from an experiment that studied the behavior of liquid crystal mixtures. Simple exploratory data analysis indicates that the regression surface has a clear ridge, and hence the piecewise linear structure of our Triogram models is ideal for this problem. The third regression example arises in the manufacture of integrated circuits and requires a slight modification of our adaptive routines that allows us to "borrow strength" between various measurements and construct a common triangulation for a suite of functionality tests. In our final application we estimate a series of bivariate densities encountered in the so-called protein-folding problem. In this case the data are naturally restricted to a triangle, suggesting that Triogram models are appropriate.

In each example we present contour and perspective plots of our Triogram fits. Unfortunately, static displays of these piecewise linear models have their limitations. An interactive environment for rotating the faceted surfaces of a Triogram model provides the best possible format for understanding these models. The authors can provide various programs implementing this type of visualization.

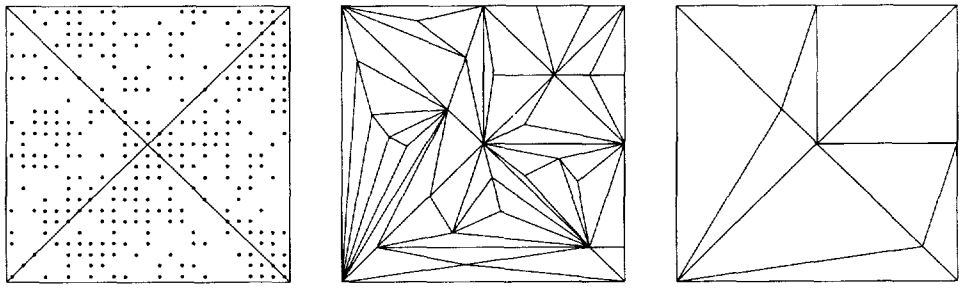
4.1 Simulated Data

Our first example involves data simulated from a bivariate regression model proposed by Gu, Bates, Chen, and Wahba (1990). The design consists of 300 "semirandom" points $\mathbf{x}_i = (x_{1i}, x_{2i})$ in the unit square. At each point $\mathbf{x}_i$ our response is $y_i = f(\mathbf{x}_i) + \varepsilon_i$, where the true regression function f is given by

$$f(\mathbf{x}) = \frac{40\exp\{8[(x_1 - .5)^2 + (x_2 - .5)^2]\}}{\exp\{8[(x_1 - .2)^2 + (x_2 - .7)^2]\} + \exp\{8[(x_1 - .7)^2 + (x_2 - .2)^2]\}}.$$

and $\varepsilon_i, i = 1, \ldots, 300$, are independent, standard normal random variables. This problem has been considered by a number of authors for evaluating the performance of various schemes based on tensor-product splines (Breiman 1991; Friedman 1991).

In the computations reported here we used the same design points as used by Gu et al. (1990). For our initial triangulation Δ_0, we divide the unit square into four triangles by drawing in both diagonals, yielding an initial model with 5 df. Figure 11a presents both the design points and Δ_0. (In Figure 11a, b, and c the point (1, 1) corresponds to the bot-



(a) (b) (c)

Figure 11. Initial Triangulation (a), Largest Triangulation (b), and Final Triangulation (c) for the Simulated Example.

tom left corner.) Because this dataset is fairly small, it is computationally feasible to fit models with many triangles and to consider many possible candidate vertices. With this in mind, we set $K = 5$ in (12) and entertain new vertices at the points given in the right hand panel of Figure 7. The maximum number of vertices was set equal to 35, although this number was rarely reached in our simulations, because we required a minimum number of four data points in each triangle. The penalty parameter a in the GCV criterion (11) was set equal to 4. Although this choice seemed to result in the smallest mean integrated squared error (MISE) across our simulations, taking a in a neighborhood of 4 yielded very similar results.

Figure 12 displays both the underlying regression function f and a Triogram fit $\hat{\phi}$ to this data. As was the case with the plots in the previous figure, the bottom corners in Figure 12a and b correspond to the point (1, 1). For this example, the largest triangulation (32 vertices) encountered during the addition phase is given in the center plot of Figure 11b, whereas the triangulation associated with the best model (nine vertices) selected by GCV is given in Figure 11c. This process was repeated 30 times, using the same semi-random design points $\mathbf{x}_i, i = 1, \ldots, 300$. The average MISE over these simulations was .139. The average mean squared error over the design points was .129, and the average number of vertices in the model selected by GCV was 9.3. The example shown in Figures 11 and 12 corresponds to the dataset with the median MISE (.138) among these 30 simulations.

When we compare our estimate to those presented by Breiman (1991) and Friedman (1991) we notice that the Triogram fit does not seem to have any of the local ridges and extrema that are evident in figure 11 of Friedman (1991). On the other hand, because the Triogram model is locally planar, it has difficulties accurately approximating such a smooth surface. To demonstrate this, we applied the Triogram algorithm to the true function without noise. For these data, Triogram selected a model with 24 vertices and a MISE of .019, whereas the Triogram model with nine vertices, like the model in Figures 11 and 12, had a MISE of .065. Thus a substantial fraction of the MISE of .139 of the Triogram procedure may be attributable to the fact that a piecewise linear surface with a moderate number of pieces does not provide an accurate approximation to a very smooth function. Presumably, Triograms using higher-order polynomials (see the discussion in Sec. 2) would do a better job.

Finally, we repeated the aforementioned computations starting from a smaller initial triangulation. For these simulations, Δ_0 consisted of the two triangles formed by either dividing the unit square along the diagonal with slope equal to 1 or -1. The results were almost identical to those reported earlier, because the first vertex added during these new simulations was usually at the point $x_1 = x_2 = .5$, essentially returning us to the starting configuration used in our initial experiments.

4.2 A Regression Surface With a Ridge

Consider again the Triogram example given in Section 1. Because the complete experiment involved just 47 data points, we selected the smallest initial model possible, a single triangle. To be more precise, Δ_0 was taken to be a 15% enlargement of the smallest triangle that contained all of the data. We obtained the 15% expansion by positioning the barycenter of the original triangle at the origin, multiplying the shifted coordinates by 1.15, and then moving the triangle back to its original position. Figure 1a shows this triangle together with the data points. As in the previous example we required the minimum number of data points in each triangle to be four. As mentioned in Section 1, subject to this constraint, the maximal model encountered during the addition phase consisted of just nine vertices. In addition, because this dataset is so small, it seemed reasonable to consider a somewhat smaller number of possible new vertices than in the previous simulated example, and so we set $K = 4$.

The Rao surface introduced in Section 3 is a useful diagnostic for uncovering structure in this data. Figure 13 evaluates the Rao statistic associated with adding a vertex at the points (12) for $K = 20$ and connects the points with a continuous piecewise linear surface. (Recall that in the regression context, the Rao statistic is simply the amount

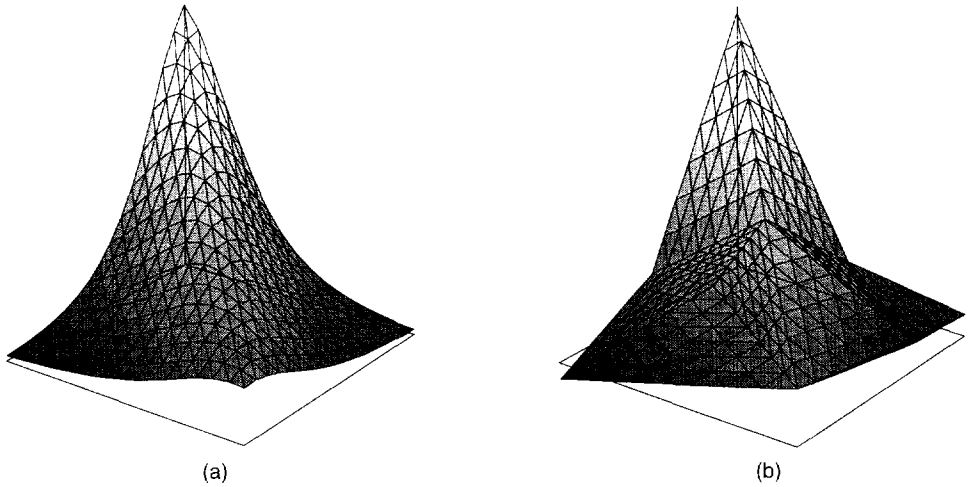


Figure 12. True Function (a) and Triogram Fit (b) for the Simulated Example.

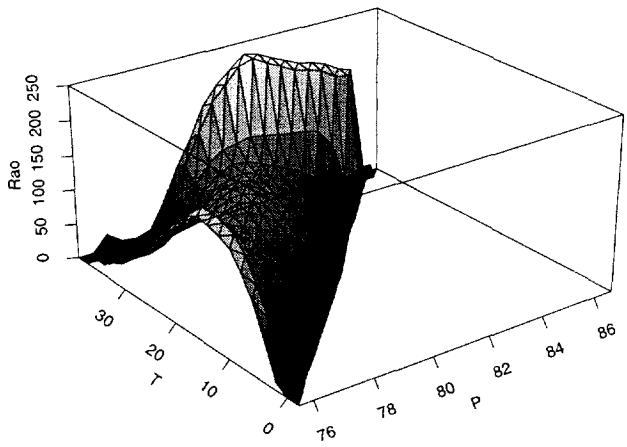


Figure 13. Rao Statistics for the First Added Vertex for the Crystal Data.

by which the residual sum of squares drops after the addition of a new basis function.) Notice that the Rao surface is fairly constant near its maximum in a strip along the edge corresponding to $T = 40$, and it drops considerably when the potential new vertex is moved to the interior of the triangle. It seems to make little difference whether we locate the first new vertex on this edge or close to this edge, because the data are sparse and the edge in question is a boundary of the initial triangulation. As mentioned earlier, Cleveland and Fuentes (1996) fitted two "hinged" planes and thus one interior edge to this data. They found that the piecewise planar model having a break along the line $T = -334.5 + 4.5P$ is optimal in the sense that it has the smallest residual sum of squares among all such single-hinged fits. This break corresponds to the dashed line appearing in Figure 1c. The Triogram algorithm places an edge in almost the same location, and in fact if we follow the more aggressive deletion scheme outlined in Section 3, we can obtain a model very similar to that derived by Cleveland and Fuentes (1996).

Although the true surface for the artificial regression function from Gu et al. (1990) is better approximated by cubic splines and their tensor products than by Triograms, the significant features in a regression surface like the one considered here should be more easily captured by the piecewise linear character of a Triogram fit. To examine this further, we conducted a small simulation study. Figure 14 presents five triangulations corresponding to a set of continuous piecewise linear functions. In each example the functions take on the value 0 at all but one vertex. The values at these remaining vertices are given explicitly in Figure 14. We evaluated each surface at 50, 200, and 1,000 randomly sampled points inside the triangle and added standard normal errors to the regression surface. Both the height of the examples in Figure 14 and the variance of the errors were such that the signal-to-noise ratio was approximately the same as in the data from Cleveland and Fuentes (1996). We repeated this process 25 times, giving us a total of 75 datasets on which we can compare the performance of Triograms to other popular surface-fitting routines.

Although each function in Figure 14 is a Triogram model, the first and third triangulations also correspond to (piecewise linear) MARS models (Friedman 1991). To make more realistic comparisons, we have placed the vertices in each of these examples so that the Triogram algorithm with $K = 4$ would not consider the correct vertex locations in its initial addition phase. For $n = 50$, we fitted models with at most 10 vertices and at least four data points in each triangle, mimicking the situation for the voltage data; for $n = 200$, we fitted models with at most 15 vertices and at least seven data points in each triangle; and for $n = 1{,}000$, we fitted models with at most 20 vertices and at least 10 data points in each triangle.

We computed the MISE over the 25 simulations for fits from Triogram, MARS (Friedman 1991), and Pimple (Breiman 1991); the results are summarized in Table 1. The typical standard errors of the estimates in Table 1 are 10–20% of the estimates themselves, for all models, sample sizes, and methods. From Table 1 we see that Triogram outperforms MARS and Pimple considerably on models 2, 4, and 5 for all sample sizes. For model 3, MARS has an edge, whereas for model 1 MARS wins for $n = 50$ and Triogram wins for $n = 1{,}000$. We should keep in mind that for models 1 and 3, MARS can pick the "correct" model in one step, whereas several steps would be required for Triogram, as the correct vertices are not in the initial search set. When we reran model 1 with $K = 5$, so that the correct vertex was in the initial search set, the MISE for Triogram was reduced by 50%, so that MARS was outperformed for all sample sizes. It is surprising how much difficulty MARS and Pimple have with model 5, even when $n = 1{,}000$. In this context Triogram models are clearly more natural than

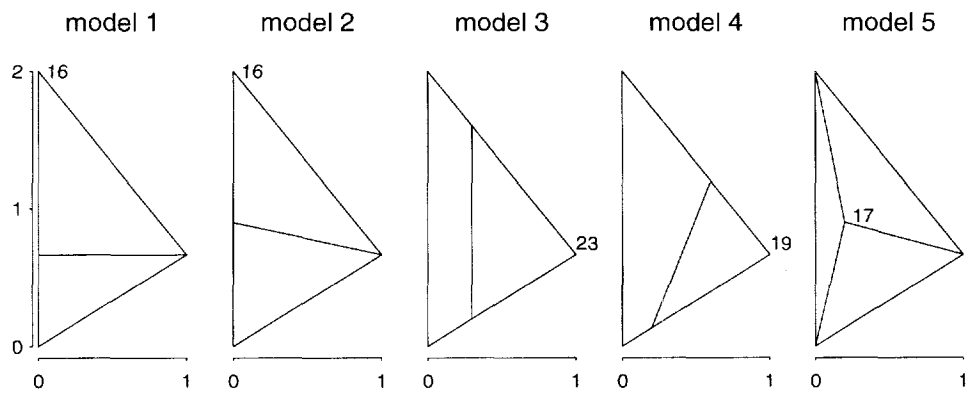


Figure 14. Five True Regression Models for a Simulation Study.

Table 1. Mean Integrated Squared Error (25 Simulations)

	n	*Model 1*	*Model 2*	*Model 3*	*Model 4*	*Model 5*
Triogram	50	.1649	.1706	.6938	.2707	.6662
Pimple[a]	50	.2347	.4101	.2348	.8172	3.0816
MARS[b]	50	.1098	.4192	.2439	1.0294	2.7530
Triogram	200	.0447	.0639	.1673	.0232	.0709
Pimple	200	.0654	.1457	.0805	.1877	.6124
MARS	200	.0436	.1363	.0665	.2658	.7242
Triogram	1,000	.0081	.0112	.0299	.0090	.0227
Pimple	1,000	.0269	.0359	.0336	.0588	.2587
MARS	1,000	.0103	.0383	.0066	.0806	.3207

[a] Excluded one simulation for model 4 with MISE of 11.0 and one simulation for model 5 with MISE of 36.6.
[b] Excluded one simulation for model 5 with MISE of 43.6.

MARS, Pimple, and smoothing spline estimates and have superior MISE performance. Ultimately, the piecewise linear character of our Triogram models is either a blessing or a curse, depending on the smoothness of the underlying functions. Clearly, each methodology has it strengths and its weaknesses. We feel that these five examples and the simulated regression problem of Gu et al. (1990) demonstrate that the Triogram models reliably capture the major features even in smooth models, and that their true advantage is in capturing ridges in the data.

4.3 A Surface Estimation Problem from the Manufacturing of Integrated Circuits

The manufacture of integrated circuits (ICs) is a complex and costly process involving hundreds of separate steps and lasting up to 12 weeks. Several hundred ICs, or chips, are fabricated simultaneously on a wafer, and the wafers themselves are processed together in groups called lots. Figure 15a presents a diagram of the location of the good and bad devices on a single wafer. The black squares denote ICs or chips that have failed one of a number functionality tests; the white squares represent good chips that will be cut from the wafer, mounted, and sold. After the fabrication process is complete, besides testing the individual devices, measurements are also taken on test structures in the "streets" or gaps between the chips. (For simplicity, we have not separated the chips in our graphical summary in Fig. 15.) Each type of test structure is repeated many times across the wafer, so that the collection of measurements corresponding to a given structure forms a surface over the wafer. Typically, these measurements are highly correlated and exhibit many similar patterns. By relating the shapes of these surfaces to the maps of defective devices on the wafer, we can learn a great deal about the manufacturing process.

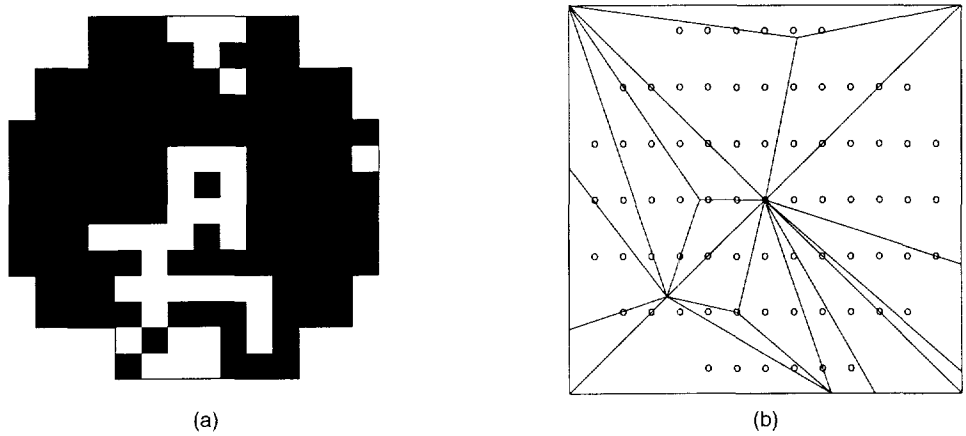
Toward this end, we use a variant of the Triogram methodology in the regression context to smooth this data, isolating common patterns. We consider 16 sets of measurements, each set consisting of observations from 73 different test sites. Because we know that there most likely are many similar shapes we will not model the various sets of measurements individually, but will instead combine the data when performing the stepwise addition procedure. Let $(V_{ik}), k = 1, \ldots, 16$ denote the (normalized) test results at the point $\mathbf{U}_i, i = 1, \ldots, 73$ and consider models of the form

$$\hat{\phi}_k(\mathbf{u}) = \sum_{j=1}^{J} \beta_{jk} B_j(\mathbf{u}) \in G,$$

where $B_j(\mathbf{u}), j = 1, \ldots, J$ is the Triogram basis of the space G associated with a given triangulation Δ. The estimates $\hat{\phi}_k$ are obtained by minimizing the pooled residual sum of squares

$$\sum_{k=1}^{16} \sum_{i=1}^{73} (V_{ik} - g_k(\mathbf{U}_i))^2, \qquad g_k \in G, k = 1, \ldots, 16. \quad (15)$$

Now, given an initial triangulation and a set of candidate vertices, we add the vertex that produces the greatest drop in the combined residual sum of squares (15). This procedure corresponds to adding new vertices based on the average Rao statistic for the different models. Once a maximum model is obtained, we perform stepwise deletion on each model separately, arriving at 16 final models, each having an underlying triangulation that is a subset of the largest model derived by the group addition procedure. Figure 15b displays the positions of the 73 test structures at which the



(a) (b)

Figure 15. The Pattern of Defective ICs on a Single Wafer (a) and the Location of Test Structures on the Wafer and the Final Triangulation Obtained by Our Combined Stepwise Addition Procedure (b). In (a), black squares denote failed devices and white squares represent chips that meet specifications and can be sold.

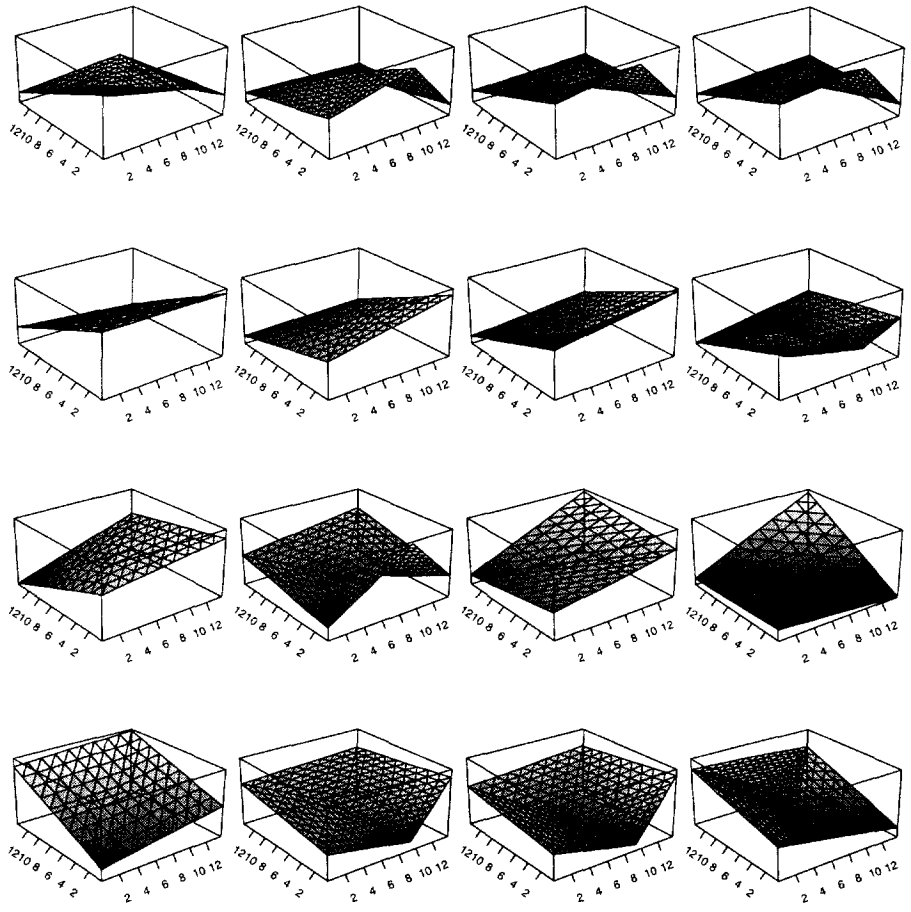


Figure 16. Sixteen Final Fits.

16 sets of measurements were taken. Starting with an initial triangulation consisting of the smallest box and both diagonals containing the data, we added vertices until a maximal model consisting of 15 basis functions was obtained. This is the triangulation in Figure 15b. The 16 final fits are given in Figure 16. (We manually ordered the test so that Triogram fits that look alike are adjacent.)

From Figure 16, we see that indeed some of the tests provide almost identical information. The Triogram fits for the middle two tests in the bottom row are almost identical to each other, and the same holds for the second, third, and fourth tests in the top row. Many of the tests seem unrelated to the patterns of failing and working devices displayed in Figure 15a. The almost planar patterns of most Triogram fits in the middle two rows, for example, may very well be related to some important characteristics of the production process, but they seem to contain no information whether the ICs are operating properly. On the other hand, the middle two Triogram fits in the bottom row have their lowest values, and the second, third, and fourth Triogram fits in the top row have their highest values in the middle front, which is the region of the wafer that contains most of the ICs that operate properly. More systematic modeling procedures can be used to perform this type of analysis, but we present these examples to illustrate how the Triogram procedure can be used as a tool for exploratory data analysis in this context.

4.4 Estimating an Unknown Density Function

The top row of Figure 17 presents three datasets that are natural candidates for Triogram density estimation. The points in these plots represent a collection of amino acids obtained from 100 protein structures taken from the Brookhaven Protein Data Bank (Hobohm, Scharf, Schneider, and Sander 1992). To characterize the local environment of each amino acid within a given protein structure, three pieces of information were recorded: the local context of the protein at the given amino acid (e.g., whether the protein is twisting around a helix), the fraction of the amino acid side-chain area buried in the protein structure, and the fraction of the side-chain area covered by polar atoms. Because the unburied portion of the amino acid is exposed to a polar solvent, the final two quantities are restricted to the upper triangle of the unit square. The plots in the top row of Figure 17 correspond to data collected from the amino acid lysine found in a helix, a coil, and a sheet.

Bivariate density estimates computed for each amino acid and each local protein structure are the basis for an approach to solving the so-called inverse folding problem (Bowie, Luthy, and Eisenberg 1991; Zhang and Eisenberg 1994). Evaluating the structure of a given protein is extremely difficult. Fortunately, determining the sequence of amino acids that comprise the protein is relatively simple. Thus it would seem reasonable to attempt to infer the protein's structure from its amino acid sequence. Unfortu-

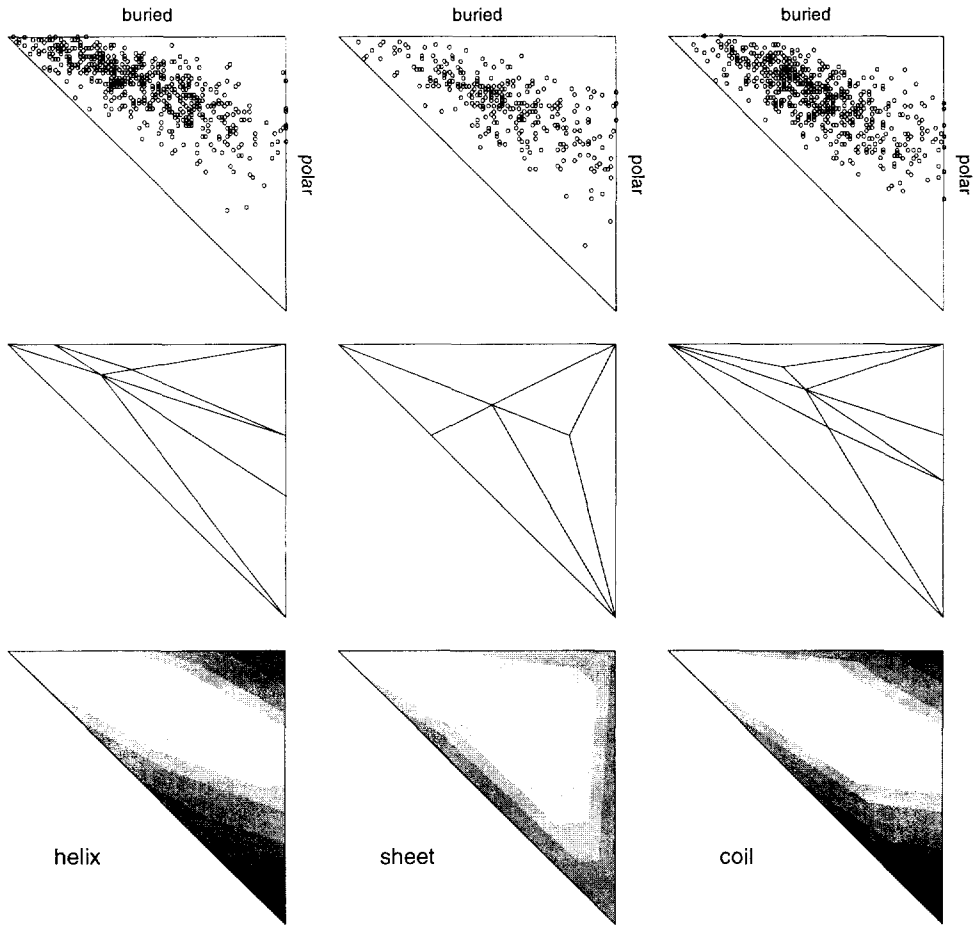


Figure 17. Triogram Density Estimates. Separate density estimates are fit for the three local protein contexts (591 amino acids found in a helix, 341 in sheets, and 593 in coils). In the bottom row the separate log-densities are contoured.

nately, many rather different sequences produce very similar structures, so the objective of the inverse folding problem is to determine which amino acid sequences might result in a given known structure. This can be accomplished by studying the propensity for certain amino acids to occur in certain local environments in a large collection of known protein structures. The procedure described by Zhang and Eisenberg involves a log-odds calculations, the main ingredient of which is a set of bivariate density estimates for the type of data given in the top row of Figure 17.

Along the top row of Figure 17 are three data clouds, one corresponding to each local context. There are 591 points in the first plot, 341 points in the second plot, and 593 points in the third plot. We first applied the Triogram procedure separately to each dataset corresponding to the three different local environments. At each step in the addition process, the set of candidate vertices consisted of the points with barycentric coordinates given in (12) with $K = 5$ relative to each of the triangles in the current triangulation Δ. We did not enforce shape restrictions on the updated triangulation when choosing between the candidates, but did insist that each triangle must contain at least 25 points. After the deletion phase, we selected a final model using BIC. In each case, the best fits were encountered during stepwise deletion. The underlying triangulations for these final models are plotted in the middle row of Figure 17, with contour plots of the corresponding log densities given in the last row of the same figure. Although the piecewise linear character of our Triogram models makes these plots somewhat jagged, they are clearly capturing the essential features of the data.

As mentioned earlier, one approach to the inverse folding problem involves a log-odds calculation based on these estimated densities. With this in mind, it is advantageous to have each of the underlying triangulations nested in some larger triangulation, and in fact it might be possible to stabilize the adaptation process somewhat by considering all three datasets simultaneously. For a given triangulation Δ, let G denote the associated space of continuous piecewise linear functions. Next, let $\mathbf{U}_{ic}, i = 1, \ldots, n_c$, denote the observations associated with local environment $c \in \{\text{helix, sheet, coil}\}$, and, as in Section 3.1, let $l_c(\boldsymbol{\beta}_c)$ denote the log-likelihood of these observations as a function of the coefficients $\boldsymbol{\beta}_c$ corresponding to the Triogram basis constructed on Δ. During the stepwise addition phase of our model building, we now compute Rao statistics using the likelihood

$$l(\boldsymbol{\beta}_{\text{helix}}, \boldsymbol{\beta}_{\text{sheet}}, \boldsymbol{\beta}_{\text{coil}}) = l_{\text{helix}}(\boldsymbol{\beta}_{\text{helix}}) + l_{\text{sheet}}(\boldsymbol{\beta}_{\text{sheet}}) + l_{\text{coil}}(\boldsymbol{\beta}_{\text{coil}}) \quad (16)$$

and add the vertex that maximizes this combined Rao statistic. Restrictions on the shape of the resulting triangulations

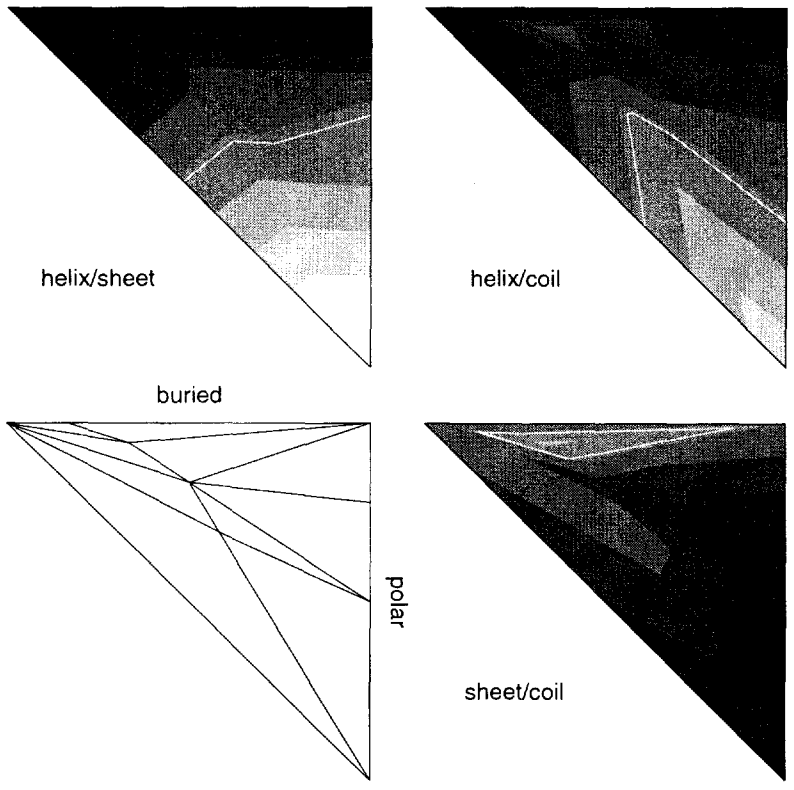


Figure 18. Log-Odds Ratios for Lysine in the Three Contexts: Helix, Sheet, and Coil. In each case the dark solid lines follow contours with value log(.5), and the light solid lines follow contours with value log(2).

as well as minimum data requirements can be enforced in the obvious way. In general, we believe that when similar functional forms are expected, this type of fitting can effectively pool the datasets to determine a common triangulation Δ from which to start the deletion phase.

Figure 18 presents the final triangulation as well as the log-odds ratios associated with the three different contexts for lysine. The plots are shaded so that as the color changes from black to white, the log-odds ratios vary from $-2 \approx \log .13$ to $3 \approx \log 20$. The dark and light lines intersect the surfaces at $\log .5$ and $\log 2$. For example, the difference of the log of the estimated density for helix and sheet when percent-buried is close to 0 and the percent-polar is almost 100 is seen to be approximately $\log .2$, as the left top corner of this panel is very dark gray.

Now, consider the difference between lysine in a helix and lysine in a sheet. Although the scatterplots in Figure 17 indicate that the center of the distribution for the sheet context is shifted more toward the barycenter of the triangle relative to the distribution of the data collected in a helix, Figure 18 suggests that if we want to decide whether unidentified lysine is in a helix or a sheet, the percent-polar (along the vertical axis) provides more evidence than the percent-buried, because the vertical color changes are more pronounced than the horizontal color changes. The same is essentially true if one wants to distinguish between helix and coil, but for distinguishing between sheet and coil, the percent-buried seems to be more informative.

5. RATES OF CONVERGENCE

In this section we discuss the rate of convergence corresponding to nonadaptive Triogram estimates; that is, Triogram models in which the triangulations are refined independent of a response. (Details can be found in Hansen 1994.) Suppose initially that given a random sample $(\mathbf{U}_1, V_1), \ldots, (\mathbf{U}_n, V_n)$ from the distribution of $(\mathbf{U}, V)$, we are interested in estimating the unknown regression function $\phi(\mathbf{u}) = E(V|\mathbf{U} = \mathbf{u}), \mathbf{u} \in \mathcal{U} \subset \mathbb{R}^2$. We assume that there exists a sequence of conforming triangulations $\Delta_n, n = 1, 2, \ldots$, of $\mathcal{U}$ and for each n construct the space G_n of piecewise linear functions described in Section 2. As our sample size n grows, we envision the triangles in Δ_n shrinking in size and increasing in number subject to some regularity conditions. By making these conditions precise, we gain insight into how we should constrain the adaptive Triogram procedure to guard against spurious effects. From a theoretical standpoint, it is not necessary for the triangulations Δ_n to be nested, and hence Δ_{n+1} need not be a refinement of Δ_n.

5.1 The Distribution of (U, V)

We assume that the density of $\mathbf{U}$ is bounded away from 0 and infinity on $\mathcal{U}$, so that as $n \to \infty$ the points of $\mathbf{U}_1, \ldots, \mathbf{U}_n$ fill $\mathcal{U}$ somewhat regularly. We must also insist that the conditional variance $\text{var}(V|\mathbf{U} = \mathbf{u}), \mathbf{u} \in \mathcal{U}$ be bounded.

5.2 Enlarging Δ_n

As mentioned in Section 2, the basis that we have chosen for G_n is well known in the finite element literature. An important property of this basis is that the L_2 norm of any function in G_n is equivalent to the l_2 norm of its coefficients

provided that there exists a constant M such that for each $\delta \in \Delta_n, n = 1, 2, \ldots$ there exists a ball $B_\delta \subset \delta$ such that

$$(\text{diam } \delta)^2/(\text{vol } B_\delta) < M.$$

We use this equivalence to demonstrate that with probability tending to 1, the empirical norm $\|g\|_n^2 = \sum_{i=1}^n g^2(\mathbf{U}_i)$ is close to its theoretical counterpart $\|g\|^2 = Eg^2(\mathbf{U})$ for all $g \in G_n$. Next, set $\bar{h}_n = \max\{\text{diam } \delta\colon \delta \in \Delta_n\}$ and $\underline{h}_n = \min\{\text{diam } \delta\colon \delta \in \Delta_n\}$ and assume that Δ_n increases with sample size so that as $n \to \infty, \bar{h}_n \to 0$, while $\log \underline{h}_n^2 + n\underline{h}_n^2 \to \infty$. When combined with the stability requirement just mentioned, this condition is sufficient to guarantee that with probability tending to 1, the space G_n is identifiable or, equivalently, the design matrix corresponding to G_n, has full rank.

5.3 Approximation Rate

For sufficiently smooth functions f, a simple Taylor expansion can be used to demonstrate that as $n \to \infty$,

$$\inf_{p \in G_n} \|p - f\|_{L_\infty(\mathcal{U})} = O(\bar{h}_n^2),$$

subject to the refinement conditions noted earlier.

Under these assumptions, the rate at which $\hat{\phi}_n$, the maximum likelihood estimate in G_n, approaches ϕ^* was derived by Hansen (1994). In the simple case described so far, if we assume that the unknown function ϕ^* has two continuous derivatives, then

$$\|\hat{\phi}_n - \phi^*\|^2 = O_P\left(\bar{h}_n^4 + \frac{\underline{h}_n^{-2}}{n}\right).$$

The first term on the right in this expression is the square of the approximation rate obtainable from G_n. Written in this way, we see precisely the penalty that we pay for using only piecewise linear functions. Improving the approximation rate will improve our rate of convergence, assuming that ϕ^* is sufficiently smooth. Now, as we collect more and more data, if we let the sets in our partition shrink so that $\bar{h}_n \sim \underline{h}_n \sim n^{-1/6}$, we obtain the rate

$$\|\hat{\phi}_n - \phi^*\|^2 = O_P(n^{-2/3}). \tag{17}$$

Under some mild extra conditions on the triangulations Δ_n, Hansen (1994) also derived the rate of convergence of the nonadaptive Triogram models in the case of a general extended linear model, which implies that (17) holds in the case of density estimation as well. We must be very clear that this is the rate associated with a nonadaptive version of the Triogram procedure. As is the case with the models discussed by Stone (1994) and Stone et al. (1997), theoretical rates such as these are useful in pointing out practically important methodologies and in indicating what types of regularity conditions should be imposed on the corresponding adaptive procedures.

In addition, this calculation indicates that we should expect certain limitations on the performance of our procedure simply because we are using linear splines, an effect studied extensively in Section 4. A word of caution is in order here, however. Although the basis that we are using appears suboptimal for smoothing functions like the example of Gu et al. (1990), data-driven adaptation ameliorates this effect considerably. This effect was observed in the case of bivariate interpolation by Rippa (1992a), who found that classically poor approximation spaces (long, thin triangles) were very effective when constructed in a data-dependent fashion.

Hansen (1994) also developed L_2 rates of convergence for general ANOVA models in the context of an extended linear model. Multivariate spline spaces of higher order are considered, as are functions involving more than two variables. In that context, an ANOVA decomposition is used to ameliorate the curse of dimensionality. In our simple Triogram setup, this is analogous to using Triogram to model selected two-factor interactions.

6. DISCUSSION

In this article we have introduced the Triogram method for function estimation using piecewise linear, bivariate splines based on an adaptively constructed triangulation. We have illustrated the technique for bivariate regression and log-density estimation and have indicated how we can directly apply our approach to model bivariate functions in the broader context of an extended linear model. The entire estimation procedure is invariant under affine transformations and is the most natural approach for modeling data when the domain of the predictor variables is a polygonal region in the plane. Although our examples dealt exclusively with estimating bivariate functions, the use of Triograms for modeling two-factor interactions in higher-dimensional functions is straightforward. In addition, we have demonstrated that Triograms are sufficiently flexible to capture the significant structure present in a variety of bivariate datasets taken from a number of different estimation contexts. These features set Triogram models apart from other estimation routines that depend heavily on a specific coordinate system and tend to be more sensitive to features that are oriented along one of the coordinate axes.

Because our estimates are piecewise linear, the results are rather crude, as made explicit by the rather slow convergence rate discussed in Section 5. By using higher-order polynomials, we not only smooth out our estimates, but also achieve a better convergence rate. However, smoothing out Triograms in this way is not trivial. We are currently investigating techniques based on the generalized vertex splines of Chui and He (1990). Essentially, by subdividing the triangles in a given triangulation, we can produce a space of continuously differentiable quadratics. Besides increased computational complexity, the price that we pay for this smoothness is that the spaces generated by our simple stepwise algorithm are no longer nested.

Finally, we are investigating alternatives to the greedy, stepwise algorithm outlined in this article. In particular, by casting the Triogram procedure into a Bayesian framework, we have developed a more efficient method for exploring "promising" triangulations. Model averaging in this context corresponds to combining several Triogram fits, each associated with a different triangulation. When applied to a smooth function, averaging removes the sharp ridges inherent in a single Triogram fit. However, when the underlying surface exhibits strong features, this averaging does not

compromise adaptability; the Triogram models with high posterior probability will all tend to share the same strong features. For a general discussion these new procedures applied to spline bases in the context of an extended linear model, the interested reader is referred to the rejoinder of Stone et al. (1977) and Hansen and Kooperberg (1998).

[Received May 1996. Revised April 1997.]

REFERENCES

Bowie, J. U., Luthy, R., and Eisenberg, D. (1991), "A Method to Identify Protein Sequences That Fold Into a Known 3-Dimensional Structure," *Science*, 253, 164–170.

Breiman, L. (1991), "The II-Method for Estimating Multivariate Functions From Noisy Data," *Technometrics*, 33, 125–143.

Chambers, J. (1995), "Overview of Version 4 of S," Technical Memorandum 95-1, AT&T Bell Laboratories.

Courant, R. (1943), "Variational Methods for the Solution of Problems of Equilibrium and Vibrations," *Bulletin of the American Mathematical Society*, 49, 1–23.

Chui, C. K. (1988), *Multivariate Splines*, Philadelphia: SIAM.

Chui, C. K., and He, T. (1990), "Bivariate C^1 Quadratic Finite Elements and Vertex Splines," *Mathematics of Computation*, 54, 169–187.

Chui, C. K., and Lai, M. (1990), "Multivariate Vertex Splines and Finite Elements," *Journal of Approximation Theory*, 60, 245–343.

Cleveland, W. S., and Fuentes, M. (1996), "Multipanel Conditioning: Modeling Data From Designed Experiments," Technical Memorandum 96-1, AT&T Bell Laboratories.

Dahmen, W. (1980), "On Multivariate B-Splines," *SIAM Journal of Numerical Analysis*, 17, 179–191.

de Boor, C. (1976), "Splines as Linear Combinations of B-Splines," in *Approximation Theory II*, eds. G. G. Lorentz, C. K. Chui, and L. L. Schumaker, New York: Academic Press, pp. 1–47.

——— (1978), *A Practical Guide to Splines*, New York: Springer.

——— (1987), "B-Form Basics," in *Geometric Modeling*, ed. G. Farin, Philadelphia: SIAM, pp. 131–148.

de Boor, C., and Höllig, K. (1982), "B-Splines From Parallelepipeds," *Journal d'Analyse Mathematique*, 42, 99–115.

——— (1988), "Approximation Power of Smooth Bivariate PP Functions," *Mathematische Zeitschrift*, 197, 343–363.

de Boor, C., Höllig, K., and Riemenschneider, S. (1993), *Box Splines*, New York: Springer.

Dierckx, P., Van Leemput, S., and Vermeire, T. (1992), "Algorithms for Surface Fitting Using Powell–Sabin Splines," *IMA Journal of Numerical Analysis*, 12, 271–299.

Dyn, N., Levin, D., and Rippa, S. (1990), "Data Dependent Triangulations for Piecewise Linear Interpolation," *IMA Journal of Numerical Analysis*, 10, 137–154.

——— (1990), "Algorithms for the Construction of Data Dependent Triangulations," in *Algorithms for Approximation II*, eds. J. C. Mason and M. G. Cox, New York: Chapman and Hall, pp. 185–192.

Farin, G. (1986), "Triangular Berstein–Bézier Patches," *Computer Aided Geometric Design*, 3, 83–127.

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines" (with discussion), *The Annals of Statistics*, 19, 1–141.

George, P. L. (1991), *Automatic Mesh Generation*, New York: Wiley.

Golub, G. H., and Van Loan, C. F. (1989), *Matrix Computations* (2nd ed.), Baltimore: Johns Hopkins University Press.

Gu, C. (1993), "Smoothing Spline Density Estimation: A Dimensionless Automatic Algorithm," *Journal of the American Statistical Association*, 88, 495–504.

Gu, C., Bates, D. M., Chen, Z., and Wahba, G. (1990), "The Computation of GCV Function Through Householder Tridiagonalization With Application to the Fitting of Interaction Spline Models," *Siam Journal of Matrix Analysis*, 10, 457–480.

Hamann, B. (1994), "A Data Reduction Scheme for Triangulated Surfaces," *Computer Aided Geometric Design*, 11, 197–214.

Hamann, B., and Chen, J. (1994), "Data Point Selection for Piecewise Trilinear Approximation," *Computer Aided Geometric Design*, 11, 477–489.

Hansen, M. (1994), "Extended Linear Models, Multivariate Splines, and ANOVA," unpublished Ph.D. dissertation, University of California, Berkeley.

Hansen, M. H., and Kooperberg C. (1998) "Spline Adaptation in Extended Linear Models," Technical Memorandum, Bell Laboratories.

Hastie, T. J., and Tibshirani, R. J. (1990), *Generalized Additive Models*, London: Chapman and Hall.

Hobohm, U., Scharf, M., Schneider, R., and Sander, C. (1992), "Selection of Representative Protein Data Sets," *Protein Science*, 1, 409–417.

Karlin, S., Micchelli, C., and Rinott, Y. (1986), "Multivariate Splines: A Probabilistic Perspective," *Journal of Multivariate Analysis*, 20, 69–90.

Kooperberg, C., Bose, S., and Stone, C. J. (1997), "Polychotomous Regression," *Journal of the American Statistical Association*, in press.

Kooperberg, C., and Stone, C. J. (1992), "Logspline Density Estimation for Censored Data," *Journal of Computational and Graphical Statistics*, 1, 301–328.

Kooperberg, C., Stone, C. J., and Truong, Y. K. (1995), "Hazard Regression," *Journal of the American Statistical Association*, 90, 78–94.

Le Méhauté, A., and Lafranche, Y. (1989), "A Knot Removal Strategy for Scattered Data in $\mathbb{R}^2$," in *Mathematical Methods in CAGD*, eds. T. Lyche and L. Schumaker, San Diego: Academic Press.

——— (1992), "Knot Removal for Scattered Data," in *Curves and Surfaces in Computer Vision and Graphics II*, Proceedings of SPIE-SPSE Conference, Boston, p. 1610.

Lyche, T. (1993), "Knot Removal for Spline Curves and Surfaces," in *Approximation Theory VII*, eds. E. Cheney, C. Chui, and L. Schumaker, San Diego: Academic Press, pp. 207–226.

Quak, E., and Schumaker, L. L. (1991), "Least Squares Fitting by Linear Splines on Data Dependent Triangulations," *Curves and Surfaces*, eds. P. J. Laurent, A. Le Méhauté, and L. L. Schumaker, San Diego: Academic Press, pp. 387–390.

Rippa, S. (1992a), "Long and Thin Triangles can be Good for Linear Interpolation," *SIAM Journal on Numerical Analysis*, 29, 257–270.

——— (1992b), "Adaptive Approximation by Piecewise Linear Polynomials on Triangulations of Subsets of Scattered Data," *SIAM Journal on Scientific and Statistical Computing*, 13, 1123–1141.

Rivara, M. C. (1984), "Adaptive Multigrid Software for the Finite Element Method," Ph.D. dissertation, K. U. Leuven, Belgium.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *The Annals of Statistics*, 6, 461–464.

Schwarz, H. R. (1988), *Finite Element Methods*, New York: Academic Press.

Stone, C. J. (1994), "The Use of Polynomial Splines and Their Tensor Products in Multivariate Function Estimation" (with discussion), *The Annals of Statistics*, 22, 118–184.

Stone, C. J., Hansen, M., Kooperberg, C., and Truong, Y. K. (1997), "Polynomial Splines and Their Tensor Products in Extended Linear Modeling," *The Annals of Statistics*, 25, 1371–1470.

Zhang, K., and Eisenberg, D. (1994), "The Three-Dimensional Profile Method Using Residue Preference as a Continuous Function of Residue Environment," *Protein Science*, 3, 687–695.