Statistical Modeling with Spline Functions Methodology and Theory

Mark H. Hansen
University of California at Los Angeles

Jianhua Z. Huang
University of Pennsylvania

Charles Kooperberg
Fred Hutchinson Cancer Research Center

Charles J. Stone
University of California at Berkeley

Young K. Truong
University of North Carolina at Chapel Hill

January 5, 2006

9
Multivariate Splines

Really early on there should be a reference to Hansen, Kooperberg, and Sardy (1998).

9.1 Preliminaries

In Chapter 3, we made use of tensor products to describe functions of more than one variable. This idea was motivated as an extension of classical $d-$way analysis of variance (ANOVA) models. The dependence of a function f on collections of the input variables could be separated in terms of *main effects* and *interactions*. At the heart of our analysis was the functional ANOVA decomposition

$$\begin{aligned} f(x_1, x_2, x_3, \ldots) = f_0 + f_1(x_1) + f_2(x_2) + f_3(x_3) + \cdots \\ + f_{12}(x_1, x_2) + f_{13}(x_1, x_3) + f_{23}(x_2, x_3) + \cdots \\ + f_{123}(x_1, x_2, x_3) + \cdots , \end{aligned} \tag{9.1.1}$$

where certain orthogonality constraints need to be applied to make this expansion identifiable. When estimating f, we typically truncated (9.1.1) to include only the main effects (an additive model) or perhaps only interactions that involve two or fewer variables. While an estimate of the complete expansion could require tremendous amounts of data, such restrictions make it possible to identify the important structural aspects of f

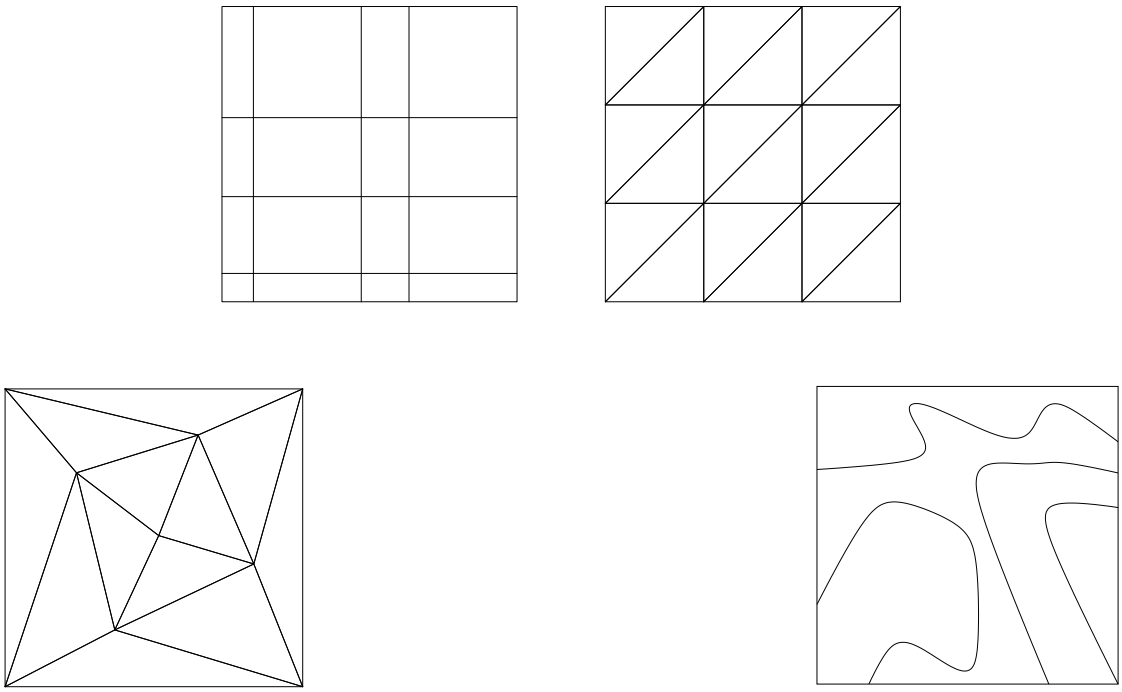

FIGURE 9.1. Four ways to specify piecewise polynomials.

in problems with even modestly sized training sets. The natural estimators in this context involve tensor products of univariate spline spaces. From a methodological perspective, the use of tensor products is consistent with the common statistical practice of generating interactions in classical regression models. As an example, let $\mathcal{G}_1$ be a spline space in the variable x_1 having basis $B_{11}, \ldots, B_{J_1 1}$ and let $\mathcal{G}_2$ be a spline space in x_2 with basis $B_{12}, \ldots, B_{J_2 2}$. Then, the two-factor interaction between x_1 and x_2 can be written as

$$\hat{f}_{12}(x_1, x_2) = \sum_i \sum_j \alpha_{ij} B_{i1}(x_1) B_{i2}(x_2)\,, \tag{9.1.2}$$

a sum of products of the basis functions of $\mathcal{G}_1$ and $\mathcal{G}_2$.

In this chapter, we take a different approach. We begin by viewing each separate element in (9.1.1) as a "surface," and consider the performance of maximum likelihood estimates taken from various flexible linear spaces. For example, as tools for building surfaces, tensor product spline spaces inherit structure from their constituent univariate components. Consider the expression in (9.1.2) for the interaction between x_1 and x_2. Suppose $\mathcal{G}_1$ has knots $t_{11}, \ldots, t_{1m_1}$ located in the range of x_1, and let $\mathcal{G}_2$ have knots $t_{21}, \ldots, t_{2m_2}$ positioned along the x_2-axis. As a result, the partial derivative

of the surface $\hat{f}_{12}$ in the variable x_1 has breaks along the lines $x_1 = t_{1i}$ for each knot t_{1i}; similarly, the partial derivative with respect to x_2 has breaks along the lines $x_2 = t_{2i}$ for each knot t_{2i}. Viewed as a function over some region in the $x_1 \times x_2$ plane, we can define $\hat{f}_{12}$ as a piecewise polynomial, where the pieces are rectangles. If $\mathcal{G}_1$ and $\mathcal{G}_2$ each consist of splines of order k, then the polynomials in each rectangle can be written as a combination of the monomials

$$x_1^{j_1}\, x_2^{j_2}\,, \qquad 0 \le j_1 < k \text{ and } 0 \le j_2 < k\,. \tag{9.1.3}$$

In the upper leftmost panel of Figure 9.1, we present the grid lines $x_1 = t_{1i}$ and $x_2 = t_{2i}$ that form the borders of the rectangular regions.

For a 1-dimensional spline space, it was sufficient to think in terms of knot sequences, or more precisely, the intervals on which we defined separate polynomials (subject to continuity constraints). In higher dimensions, the problem becomes much more complex, as there are simply many more ways to create separate pieces. For many years, numerical analysts have studied the properties of multivariate, piecewise-polynomials. One popular class of such models involve grids formed from collections of triangles. Consider, for example, the upper rightmost panel in Figure 9.1. Here, we again view the lines running through the square region as defining segments along which the piecewise polynomial surface is allowed to be discontinuous (or have discontinuous partial derivatives in some direction); and within each triangle, we assume the surface to be a (multivariate) polynomial. The grid in this panel is often referred to as a 3-directional mesh because in addition to the horizontal and vertical grid lines, there are breaks along the lines with unit slope (in the same way, one might read that the tensor product grid is a 2-directional mesh).

In addition to entertaining different grids, we can also consider alternative polynomial specifications within each cell. The monomials given in (9.1.3) are said to have *coordinate order k* because the highest order of any individual component is k. For $k = 2$, 3 and 4, these spaces are known in the engineering literature as bilinear, biquadratic and bicubic polynomials, respectively. As we can see from (9.1.2), they arise naturally when modeling with tensor products. When triangles are used to define the mesh, however, it is typical to specify the polynomial pieces as combinations of the monomials

$$x_1^{j_1}\, x_2^{j_2}\,, \qquad 0 \le j_1 + j_2 < k\,, \tag{9.1.4}$$

for some order k. These terms are said to have *coordinate order k*. By way of comparison, for $k = 3$, the spaces (9.1.3) and (9.1.4) differ only in the monomial $x_1^2 x_2^2$. While the classical Taylor's expansion for smooth, multidimensional functions involves tensor-product or coordinate order polynomials, a more general result covers the approximation power of total order spaces as well. In general, the approximation error achievable by a

polynomial in a neighborhood δ is given by

$$(\text{size of } \delta) \times (\text{size of high-order derivatives of } f)\,.$$

For the bivariate space (9.1.3), the term on the left consists of the partial derivatives

$$\frac{\partial^k f}{\partial x_1}, \quad \text{and} \quad \frac{\partial^k f}{\partial x_2}$$

while for (9.1.4) we need a bound on

$$\frac{\partial^{i_1}}{\partial x_1}\frac{\partial^{i_2}}{\partial x_2} f \qquad \text{where } i_1 + i_2 = k\,.$$

The important point here is that there are numerous ways to specify the ingredients necessary to define a spline space in several variables, including what we mean by a multivariate polynomial. Readers interested in details of this discussion are referred to Chapter 13 of Schumaker (1981).

As we have seen, the smoothness properties (the number of continuous partial derivatives) of tensor product models can be derived easily from the separate univariate spaces. An alternate recipe for constructing multivariate splines starts with the mesh and the order of the separate polynomial pieces. Then comes the difficult task of enforcing smoothness constraints across the boundaries of each piece. The literature on finite element methods is rich with such constructions over very regular triangular and rectangular meshes.[1] Spaces even exist for mixed grids consisting of both types of cell (Schwartz 1981). In some cases, it is possible to define a locally supported basis, reminiscent of the B-splines. For the 3-directional mesh, for example, the space of so-called box splines can be thought of as a multivariate version of the cardinal B-splines. Their support consists of a modest number of neighboring triangles depending on the order of the spline space and the smoothness desired. See de Boor, Höllig, and Riemenschneider (1993) for a complete description of box splines.

In the lower, leftmost plot of Figure 9.1 the grid consists of a much freer arrangement of triangles. Less regular meshes such as this one can be complicated to work with. Enforcing smoothness constraints can be difficult, and it is necessary to place restrictions on the number of continuous derivatives achievable relative to the order of the polynomial. Typically, for the spaces to not degenerate to a single polynomial or otherwise compromise

[1] The basic concept of finite element analysis is that a body or structure may be divided into smaller pieces of finite dimensions called as *finite elements*. Then the behavior of physical quantities in each element is described, and the separate pieces are assembled to form an approximation of the original system. When modeling quantities like stress or strain across the structure, we expect a certain degree of continuity when moving between elements. This provides us with a formal connection between these engineering tools and the concepts we have discussed so far from approximation theory.

FIGURE 9.2. Bias-variance characteristics of an OLS fit. The data have been standardized to have a signal-to-noise ratio of 2-to-1. Observe how characteristics of each spline space influences the look of the final fit.

the approximation rate of the space, the order of the splines needs to be relatively large. In such settings, forming a local basis can be extremely difficult, and the dimension of the resulting space is not entirely clear. We examine these complications for bivariate surfaces in the next section. In the final panel of Figure 9.1 we present a mesh in which the individual pieces are no longer separated by line segments. Some results for this kind of mesh exist, mainly establishing upper and lower bounds on the dimension of the space (Chui 1988). We mention the possibility of such wild structures only to give the reader an idea of what is possible. For the rest of this chapter, we will consider only meshes consisting of triangles. Still, given the difficulties involved in working with such relatively straightforward grids, one may wonder why we should stray from tensor products at all.

It should be intuitively clear that the grids given in Figure 9.1 differ in their ability to resolve features in a bivariate function (although asymptotically as long as the separate pieces shrink, they can be made to perform similarly). While we motivated the use of tensor products mainly from a statistical standpoint, it should be clear that if we are interested in estimating a surface, a more efficient representation might be possible from something other than a rectangular mesh. This suggests adapting the underlying mesh to the characteristics of the surface. The idea is a logical extension of our desire to place knots in a univariate fit near important structures. In Figure 9.2 we illustrate three bivariate functions and paired plots of squared bias and variance for the two meshes given in the top row of Figure 9.1. In each case, we are working with quadratic splines and have

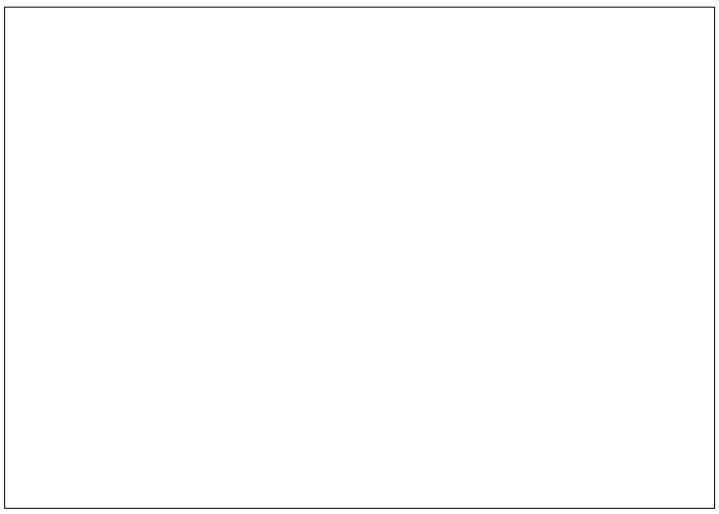

FIGURE 9.3. The data, a MARS fit and a Triogram fit.

used the same degrees of freedom for the tensor product and box spline fits. For the middle panel, the 3-directional mesh is much better able to capture the strong diagonal orientation of this surface. As we have seen in earlier chapters, adapting the structure of the spline space to characteristics of the unknown function can improve the fit.

We now explore the use of linear bivariate splines in a simple example. We focus on continuous, piecewise linear functions and in so doing, sidestep many of the issues introduced above. These simple spaces, however, provide us with sufficient flexibility to model strong features like peaks and ridges. We will introduce a simple mechanism for mesh adaptation that is easily motivated by our greedy schemes for knot addition and deletion in a univariate fit.

9.2 An application

Cleveland and Fuentes (1996) analyze data collected as part of an experiment on the processing of liquid crystal mixtures. The response is the voltage V necessary to turn a mixture 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. After extensive exploratory data analysis, Cleveland and Fuentes (1996) fit

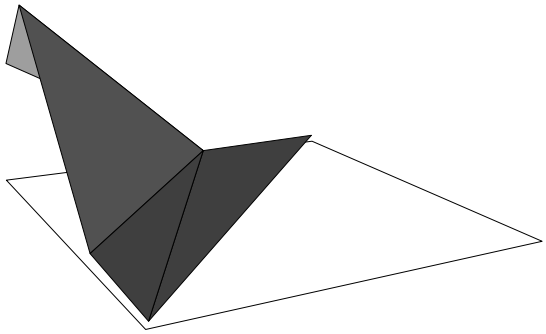

FIGURE 9.4. Final Triogram fit for the crystal data.

a model consisting of two half planes that join along a line in T and P space.

Two factors confound the usual application of a tensor product model to these data. First, a mesh with axis-oriented grid lines will not be able to efficiently represent the break observed by Cleveland and Fuentes. Next, the data distribution is restricted to a triangle, and the bias-variance lesson given above will confound a tensor product model. In In the leftmost panel of Figure 9.3, we present the data in the $T-P$ plane and indicate the break found by Cleveland and Fuentes. Note that there are only 47 data points collected by experimenters across 3 separate experiments. In the middle panel, we plot the grid associated with a MARS fit to these data. In the final panel, we present the results of applying an adaptive fitting routine that grows a triangulation in the same greedy fashion that univariate spline models added knots to a curve estimate.

Starting from a simple linear fit defined over the triangular support of the data, 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 of nine vertices, and is shown in Figure 9.3. 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 9.3. A perspective plot of this fit is given in Figure 9.4.

The largest model that was fitted had only 9 vertices, since none of the triangles could be further subdivided without violating the requirement on the minimum number of data points. The GCV criterion with penalty parameter 4 selected a Triogram model with six vertices. The largest triangulation encountered during the addition phase and the triangulation associated with the best model are shown, respectively. A perspective plot of the fit is given in the left hand panel.

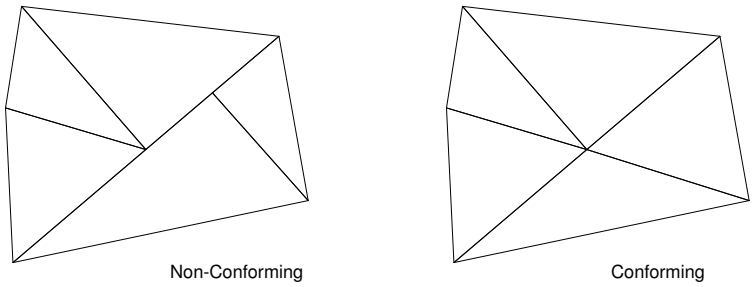

FIGURE 9.5. Non-conforming and conforming triangulations.

9.3 The methodology

9.3.1 Bivariate spline spaces

In this section, we will focus on piecewise polynomials (of some fixed total order k) defined over a mesh consisting of triangles. We begin with some basic notation about bivariate splines. Let $\mathcal{X}$ be a compact region in the plane, and let $\triangle$ be a collection of closed subsets of $\mathcal{X}$ having disjoint interiors satisfying

$$\mathcal{X} = \cup_{\delta \in \triangle} \delta \, .$$

The set $\triangle$ is said to form a *tessellation* of $\mathcal{X}$. In Figure 9.1, the region $\mathcal{X}$ is a square, and the three grids represent tesselations. If each set $\delta \in \triangle$ is a planar triangle, then $\triangle$ represents a *triangulation* of $\mathcal{X}$. A triangulation $\triangle$ is said to be *conforming* if the nonempty intersection between pairs of triangles in $\triangle$ consists of either a single shared vertex or an entire common edge. The upper rightmost and lower leftmost panels in Figure 9.1 each contain conforming triangulations. (See Figure 9.5 for a precise illustration of this concept.)

Constructing multivariate spline spaces

Following the approach in Chapter 3, it seems natural to define a spline space in terms of a mesh and the smoothness conditions that the surface should satisfy across each segment. When we move to even two dimensions, however, we are limited in what we can say about spaces of piecewise polynomials defined in this way, even if we restrict our attention to triangulations. For example, the dimension of such a space can depend on the geometry of the mesh. A simple example of this was given in Morgan and Scott (1975) and is reproduced in Figure 9.6. A space of quadratic polynomials with continuious first partial derivatives defined over the mesh in this figure is either 6 or 7, depending on whether or not the grey triangle in the center is symmetric. Complications such as this make it difficult to derive polynomial spaces that are usable in statistical applications.

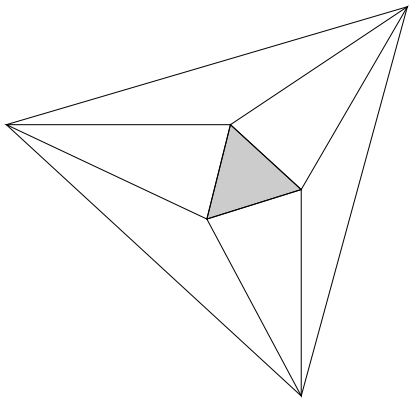

FIGURE 9.6. The dimension of the space of quadratic polynomials with continuious first partial derivatives defined over the mesh above is 6 or 7 depending on whether or not the grey triangle in the center is symmetric.

Aside from characterizing the space, we also need to derive an basis. 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 $\triangle$ and attempt to construct smooth, piecewise polynomial basis functions that have small support by enforcing smoothness conditions across the edges in $\triangle$. This so-called *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 possessing r continuous derivatives must have (total) order at least $3r+3$ (see de Boor and Höllig 1988). We can remove these restrictions by either considering only extremely regular meshes (like the 3-directional grid in Figure 9.1); or by subdividing the triangles in $\triangle$. In Figure 9.7, we present one type of subdivision that has proved useful in designing spline models. The class is known as vertex splines and was introduced in Chui and He (1990). Given this elaborate expansion of the mesh, these authors buy enough flexibility to derive explicit equations for a locally supported basis. In this case, local means having support restricted to all the triangles sharing a common vertex.

Other approaches define the mesh and the basis functions at the same time, a procedure that is analogous to "pulling apart knots" in a space of univariate B-splines. Recall from Chapter 2 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

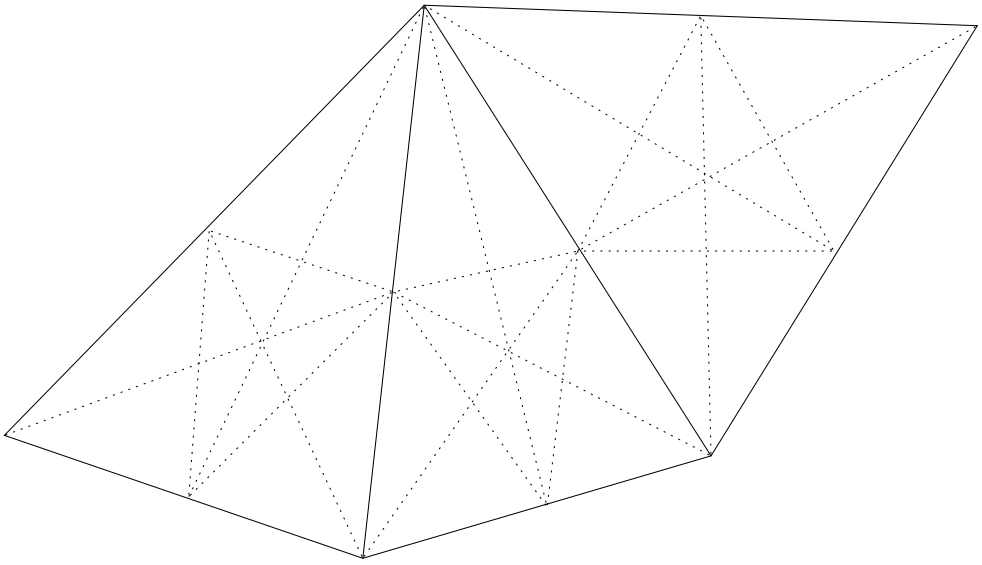

FIGURE 9.7. Introducing 12 new triangles (dashed lines) for each triangle in the original mesh (solid lines).

triangulation $\triangle$ (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 the multivariate case, 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; (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).

Complications in both the characterization and specification of arbitrary piecewise polynomials has led to the development of several important special cases, like those based on regular meshes (see the 3-directional grid given in the middle panel of Figure 9.1). Rather than limit the structure of the approximation space, we will prefer to restrict the order of the polynomial pieces. By focusing on continuous, piecewise linear functions or *linear splines*, these difficulties disappear. Throughout this chapter, we will focus mainly on surfaces, or bivariate characterizations (although the construction can be easily generalized to higher dimensions). We have also chosen linear splines because data-driven rules for adaptively choosing $\triangle$ are more easily explored in this rather simple setting. We will have more to say on

this topic when we compare the Triogram algorithm (Hansen, Kooperberg, and Sardy 1998) with well-known techniques from approximation theory.

Linear splines

Let $\mathcal{G}$ denote the space of continuous, piecewise linear functions over a given triangulation $\triangle$: Each $g \in \mathcal{G}$ is continuous on $\mathcal{X}$, and the restriction of g to $\delta \in \triangle$ is a linear function. Defined in this way, $\mathcal{G}$ is a finite dimensional, linear space and there is a natural association between the vertices $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_J$ of the triangles in $\triangle$ and a set of basis functions $B_1(\boldsymbol{x}), \ldots, B_J(\boldsymbol{x})$ that span $\mathcal{G}$. Define $B_j(\boldsymbol{x})$ to be the unique function that is linear on each of the triangles in $\triangle$ and takes on the value 1 at $\boldsymbol{v}_j$ and 0 at the remaining vertices in the triangulation. This collection of tent functions was originally proposed in Courant (1943), and is frequently used in the finite element method. As we will see 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; 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 $\delta \in \triangle$ having vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$. The barycentric coordinates of any point $\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2$ are defined as a triple $\boldsymbol{\phi}(\boldsymbol{x}) = (\varphi_1(\boldsymbol{x}), \varphi_2(\boldsymbol{x}), \varphi_3(\boldsymbol{x}))$, such that

$$\boldsymbol{x} = \varphi_1(\boldsymbol{x})\boldsymbol{v}_1 + \varphi_2(\boldsymbol{x})\boldsymbol{v}_2 + \varphi_3(\boldsymbol{x})\boldsymbol{v}_3$$

and

$$\varphi_1(\boldsymbol{x}) + \varphi_2(\boldsymbol{x}) + \varphi_3(\boldsymbol{x}) = 1.$$

These conditions are equivalent to the following set of linear equations

$$\begin{pmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \varphi_1(\boldsymbol{x}) \\ \varphi_2(\boldsymbol{x}) \\ \varphi_3(\boldsymbol{x}) \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}, \tag{9.3.1}$$

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

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

As its name suggests, the absolute value of SignedArea$(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$ is just the area of the triangle with vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$. By applying Cramer's method to the set of equations (9.3.1) we find that $\varphi_1(\boldsymbol{x})$ is given by the ratio

$$\varphi_1(\boldsymbol{x}) = \varphi_1(x_1, x_2) = \frac{\text{SignedArea}(\boldsymbol{x}, \boldsymbol{v}_2, \boldsymbol{v}_3)}{\text{SignedArea}(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)}. \tag{9.3.2}$$

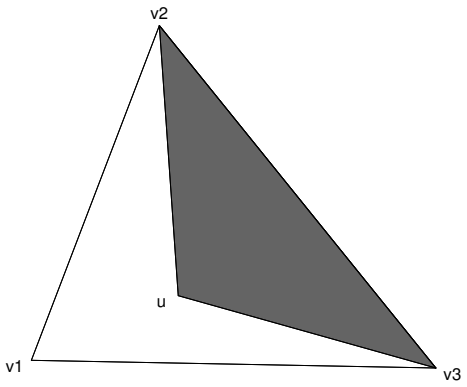

FIGURE 9.8. The barycentric coordinates of a point $\boldsymbol{x}$ relative to the triangle with vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ are expressed as ratios of signed areas. In this case, the function $\varphi_1(\boldsymbol{x})$ is the ratio SignedArea$(\boldsymbol{u}, \boldsymbol{v}_2, \boldsymbol{v}_3)$/SignedArea$(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$

This relationship is illustrated in Figure 9.8.

From the expression in (9.3.2), we see that the barycentric coordinates are linear functions of x_1 and x_2, where $\boldsymbol{x} = (x_1, x_2)$, and satisfy the interpolation conditions

$$\varphi_i(\boldsymbol{v}_j) = \begin{cases} 0 & i \neq j, \\ 1 & i = j, \end{cases} \qquad i, j = 1, 2, 3\,; \tag{9.3.3}$$

hence the vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ have barycentric coordinates $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$, respectively. Furthermore, from (9.3.2) we see that the points on the edge connecting $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ 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 (9.3.3) can be used to demonstrate that the functions $\varphi_1(\boldsymbol{x})$, $\varphi_2(\boldsymbol{x})$, and $\varphi_3(\boldsymbol{x})$ are linearly independent and hence constitute a basis of the space of linear functions of $\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2$. While it is customary in statistical applications to choose the basis comprised of the constant function 1 and the two coordinate functions x_1 and x_2, the barycentric basis has the advantage that it is invariant under affine transformations such as rotations: given any nonsingular, 2-by-2 matrix A and any vector $\boldsymbol{b} \in \mathbb{R}^2$,

$$\varphi_i(\boldsymbol{x}) = \varphi_i^*(A\boldsymbol{x} + \boldsymbol{b}), \qquad \text{for } i = 1, 2, 3 \text{ and } \boldsymbol{x} \in \mathbb{R}^2\,, \tag{9.3.4}$$

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

Returning to our triangulation $\triangle$ and the space of continuous, piecewise linear functions $\mathcal{G}$, we let $\delta \in \triangle$ be a triangle with vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$

and observe that from the interpolation conditions (9.3.3), the functions $\varphi_1(\boldsymbol{x})$, $\varphi_2(\boldsymbol{x})$ and $\varphi_3(\boldsymbol{x})$ are exactly the basis functions $B_1(\boldsymbol{x})$, $B_2(\boldsymbol{x})$ and $B_3(\boldsymbol{x})$ for $\boldsymbol{x} \in \delta$. As an immediate consequence of this construction, we find that the basis of tent functions $B_1, \ldots, B_J$ associated with the triangulation $\triangle$ are bounded between zero and one and satisfy

$$B_1(\boldsymbol{x}) + \cdots + B_J(\boldsymbol{x}) = 1, \qquad \boldsymbol{x} \in \mathcal{X}'$$

a property shared by univariate B-spline bases. From (9.3.2) we also find that for any nonsingular, 2-by-2 matrix A and any vector $\boldsymbol{b} \in \mathbb{R}^2$,

$$B_j(\boldsymbol{x}) = B_j^*(A\boldsymbol{x} + \boldsymbol{b}), \qquad \text{for all } \boldsymbol{x} \in \mathbb{R}^2 \ ,,$$

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

The price for this simplicity is that our Triogram estimates are crude. At the end of this chapter, 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 (1992) 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.

Higher-order bivariate splines

As mentioned above, 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

$$1,\ x_1,\ x_1^2,\ x_2,\ x_2^2,\ x_1x_2$$

is also spanned by the functions

$$\varphi_1^{i_1}(\boldsymbol{x})\,\varphi_2^{i_2}(\boldsymbol{x})\,\varphi_3^{i_3}(\boldsymbol{x}) \qquad \text{for } i_1 + i_2 + i_3 = 2\,, \tag{9.3.5}$$

where $\boldsymbol{x} = (x_1, x_2)$ 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 (9.3.4). When moving from a single triangle to a collection of triangles $\triangle$, 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 $\triangle$, reducing the task to a straightforward accounting problem (see Chui and Lai 1990). While this procedure is still subject to the severe conditions linking smoothness and degree, regular subdivision of $\triangle$ can also make use of the B-net structure to generate, for example, quadratic splines with continuous first partial derivatives in each coordinate direction can be defined over arbitrary triangulations (see Chui and He 1990).

9.3.2 Maximum likelihood estimation.

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

$$\hat{g} = \arg\max_{g \in \mathcal{G}} l_n(g)\,, \qquad \text{where} \qquad l_n(g) = \sum_{i=1}^{n} l(g, W_i)\,. \tag{9.3.6}$$

Equivalently, $l_n(g)$ can be written as $l_n(\boldsymbol{\theta})$ where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_J) \in \mathbb{R}^J$ and

$$g(\boldsymbol{x}) = \theta_1 B_1(\boldsymbol{x}) + \cdots + \theta_J B_J(\boldsymbol{x})\,, \qquad \boldsymbol{x} \in \mathcal{X}\,.$$

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

Consider the normal regression model. Let $\boldsymbol{W} = (\boldsymbol{X}, V)$ with $V \in \mathbb{R}$ and set $f(\boldsymbol{X}) = E(V|\boldsymbol{X})$. Then, given observations $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_n$, we estimate $\hat{g}(\cdot)$ by

$$\hat{g}(\boldsymbol{x}) = \arg\max_{g \in \mathcal{G}} \sum_{i=1}^{n} (g(\boldsymbol{X}_i) - V_i)^2\,,$$

yielding the normal equations

$$\hat{\theta}_1 \langle B_i, B_1 \rangle_n + \cdots + \hat{\theta}_J \langle B_i, B_J \rangle_n = \langle B_i, V(\cdot) \rangle_n\,, \qquad 1 \le i \le J\,, \tag{9.3.7}$$

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

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

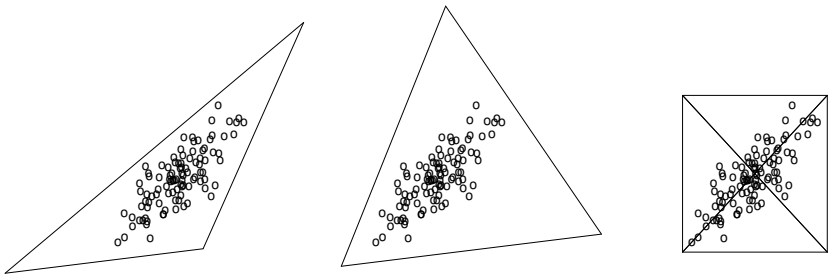

FIGURE 9.9. Three standard initial triangulations.

By construction, the ith equation in (9.3.7) involves only those coefficients $\hat{\theta}_j$ for which the vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are joined by an edge in $\triangle$. The maximum of $|i-j|$, taken over all pairs i, j such that $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are connected by an edge in $\triangle$, is referred to as the bandwidth of $\triangle$. Schwarz (1988) describes a number of well-known algorithms that renumber the vertices of an existing triangulation $\triangle$ 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 Loan 1989) to solve the normal equations (9.3.7).

So far in this section, we have considered applying maximum likelihood to fit a Triogram model only for a fixed mesh $\triangle$ (and hence a fixed space $\mathcal{G}$). In the remainder of the 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 will end this section by making connections between Triograms and similar adaptive procedures in the literature on approximation theory.

9.3.3 A stepwise algorithm

The adaptive Triogram procedure starts with an initial triangulation $\triangle_0$ and a maximum likelihood estimate $\hat{g}_0 \in \mathcal{G}_0$. In many applications a natural initial configuration may be determined by the shape of $\mathcal{X}$ or a priori knowledge about f. For situations in which the initial triangulation is not so clearly defined, there are some default alternatives: one might consider the smallest triangle, the smallest equilateral triangle, and the smallest axis-oriented rectangle that contain all the data $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, 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. In Figure 9.9 we present 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 $\hat{g}_0$ is just a plane. In general, if the initial model is not sufficiently flexible to capture the major features of the data, we enrich $\mathcal{G}_0$ by stepwise refinements to the triangles $\delta \in \triangle_0$.

During the addition phase we produce a sequence of nested spaces $\mathcal{G}_0 \subset \mathcal{G}_1 \subset \cdots \subset \mathcal{G}_m$ of continuous, piecewise linear functions having dimensions $p, p+1, \ldots, p+m$, respectively. As usual, associated with each space $\mathcal{G}_i$ is a conforming triangulation $\triangle_i$ of $\mathcal{X}$. 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 $\mathcal{G}_{i+1}$ from $\mathcal{G}_i$ involves adding a single new vertex to the underlying triangulation $\triangle_i$. There are obvious constraints on this process because the mesh $\triangle_{i+1}$ corresponding to $\mathcal{G}_{i+1}$ must also be a conforming triangulation, and $\mathcal{G}_i$ must be a subspace of $\mathcal{G}_{i+1}$. In addition, we must only make changes to $\triangle_i$ that yield a space $\mathcal{G}_{i+1}$ in which the maximum likelihood equations (9.3.6) 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 $\triangle_i$ to produce a refined triangulation $\triangle_{i+1}$ and a new space $\mathcal{G}_{i+1}$ representing a single degree-of-freedom change to $\mathcal{G}_i$. 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 f is a regression function, for example, we select the vertex that has the greatest decrease in the residual sum of squares when it is added to $\triangle_i$. 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 $\mathcal{G}'_0 \supset \mathcal{G}'_1 \supset \cdots \supset \mathcal{G}'_{m'}$, 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 $\triangle'_0$. This process is also subject to a number of constraints imposed by our requirements that $\mathcal{G}'_{i+1}$ be a subspace of $\mathcal{G}'_i$ and that the mesh associated with each space must be a conforming triangulation. Details about how vertices are identified as candidates for deletion will be given in the Section 9.3.4. For the purpose of this discussion, however, we simply assume that at each step i there are a number of vertices that can be removed from $\triangle'_i$ to produce a smaller triangulation $\triangle'_{i+1}$ and a new space $\mathcal{G}'_{i+1}$ representing a single degree of freedom change to $\mathcal{G}'_i$. We choose from among these candidates the one that minimizes the Wald statistic associated with deleting the corresponding basis element from $\mathcal{G}'_{i+1}$. For example, when f is a regression function, we select the vertex that yields

the least increase in the residual sum of squares when it is deleted from $\triangle_i'$. 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, Hansen, Kooperberg, and Truong 1997). Regression is the only estimation context for which this does not represent a computational advantage, since 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 f 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 $\widehat{l}_\nu$ denote the fitted log-likelihood for the νth model, and for a fixed penalty parameter a, set

$$\mathrm{AIC}_{a,\nu} = -2\widehat{l}_\nu + ap_\nu \tag{9.3.8}$$

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 f is a regression function we discriminate between models on the basis of their GCV score (Friedman 1991)

$$\mathrm{GCV}_{a,\nu} = \frac{1}{n}\frac{\mathrm{RSS}_\nu}{\left(1 - \frac{ap_\nu}{n}\right)^2}, \tag{9.3.9}$$

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 (9.3.9) for the number of parameters that are used in the initial model, since 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 in (Friedman 1991), so this is our default choice in the Triogram software.

In the remainder of this section 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.

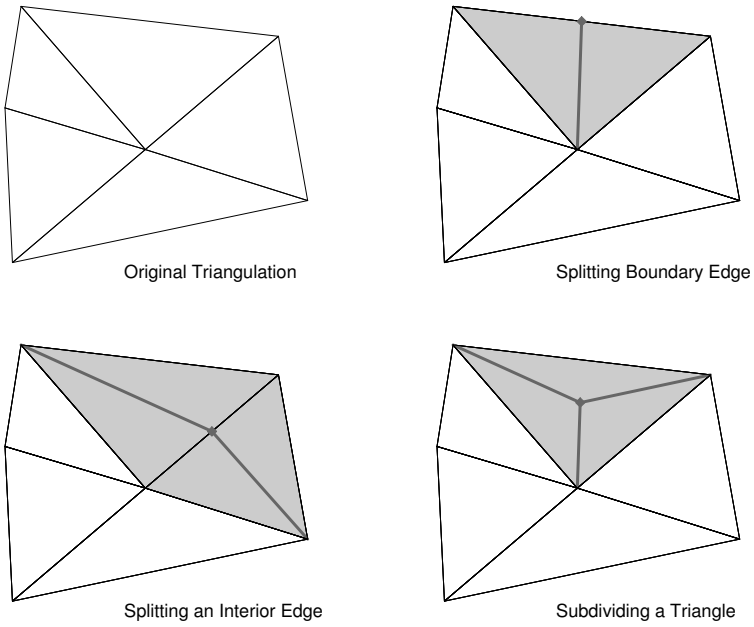

FIGURE 9.10. Three ways to add a new vertex to an existing triangulation. Each addition represents the introduction of a single basis function, the support of which is colored gray.

9.3.4 Stepwise addition

Inserting a new vertex into an existing triangulation $\triangle$ requires a rule for connecting this point to the vertices in $\triangle$ so that the new mesh is also a conforming triangulation. In Figure 9.10, we illustrate three options for vertex addition: we can place a new vertex on either a boundary or an interior edge, splitting the edge, or we an add a point to the interior of one of the triangles in $\triangle$. Note that the space obtained by adding a vertex $\boldsymbol{v}$ to an interior edge of a triangle $\delta \in \triangle$ cannot be achieved as the limit of spaces constructed by adding $\boldsymbol{v}$ to the interior of δ. In this case, if $\boldsymbol{v}$ is very close to an edge of δ 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 $\boldsymbol{v}$ 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 $\triangle$, we construct a set of candidate vertices by considering the points with barycentric coordinates

$$\left(\frac{k_1}{K+1}, \frac{k_2}{K+1}, \frac{K+1-k_1-k_2}{K+1} \right)_{\delta}, \qquad \delta \in \triangle, \tag{9.3.10}$$

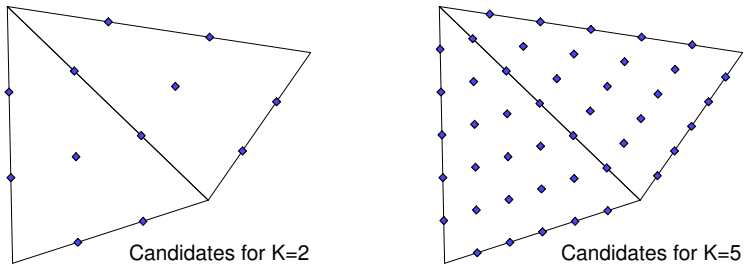

FIGURE 9.11. Candidate vertices for $K = 2$ and $K = 5$.

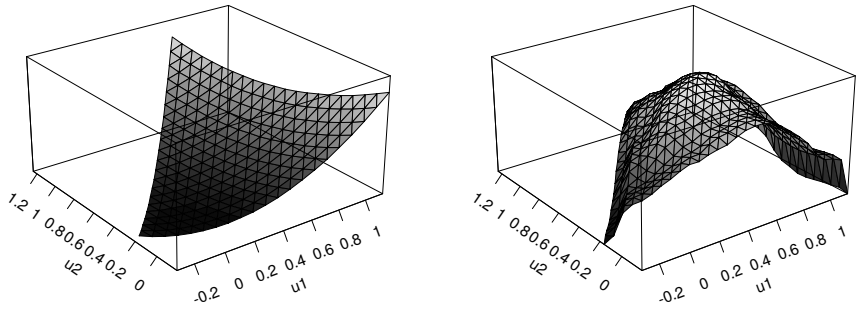

FIGURE 9.12. Rao statistics for adding a knot. The left surface is truth, and the right right is the Rao surface for adding single knot to simple linear fit.

where k_1, k_2 and K are nonnegative integers satisfying $k_1 + k_2 \leq K + 1$ and no coordinate equals one. We have introduced a subscript "δ" to make it clear that these points are calculated for each triangle in $\triangle$. The positions of the candidate knots calculated with $K = 2$ and $K = 5$ in (9.3.10) are plotted in Figure 9.11. In order to avoid the degeneracies mentioned above, we suggest modest values of K, with 5 being 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 $\triangle$ with many long, thin triangles or triangles containing little or no data tend produce highly unstable estimates. This notion is made precise at the end of the chapter 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 use-

ful in understanding both the behavior of the Triogram procedure as well as the placement of significant structures in a particular data set. In Figure 9.12, we present the Rao surface associated with adding a new vertex to a partition $\triangle$ consisting of just one triangle. In this case, we are using ordinary least squares to estimate ϕ^*, the simple quadratic $x_1^2 + x_2^2$ plotted in the left hand portion of the figure. We generated 100 points uniformly in the triangle and added independent, normal noise to ϕ^* so that the signal to noise ratio was three to one. In the panel on the right, we present the Rao surface for adding a new node to the triangle. Since we are estimating a regression function, the height of this surface at a particular point $\boldsymbol{x}$ is equivalent to the drop in the residual sum of squares when a new vertex is added to $\triangle$ at $\boldsymbol{x}$. 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 $\triangle$ form the boundary of $\mathcal{X}$ 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 $\delta \in \triangle$, we also calculate the Rao statistic at a small number of points following the recipe in (9.3.10), but fit a polynomial $\hat{p}_\delta(\boldsymbol{x})$ using the basis (9.3.5). The new vertex is then chosen as from among the points

$$\text{argmax}_{\boldsymbol{x} \in \delta}\, \hat{p}_\delta(\boldsymbol{x}) \qquad \text{for } \delta \in \triangle\,.$$

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 $B(\cdot)$, again using the barycentric coordinate functions described in Section 9.3.1. Suppose for the moment that we want to introduce a vertex $\boldsymbol{v}$ in the interior of a triangle δ with vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$. Recall that the barycentric coordinate functions $\boldsymbol{\phi}(\boldsymbol{x}) = (\varphi_1(\boldsymbol{x}), \varphi_2(\boldsymbol{x}), \varphi_3(\boldsymbol{x}))$, $\boldsymbol{x} \in \mathbb{R}^2$, associated with δ form a basis for the space of linear functions in $\boldsymbol{x} = (x_1, x_2)$. Therefore, any line in the plane can be expressed in the form

$$\alpha_1\varphi_1(\boldsymbol{x}) + \alpha_2\varphi_2(\boldsymbol{x}) + \alpha_3\varphi_3(\boldsymbol{x}) = 0\,, \quad \boldsymbol{x} \in \mathbb{R}^2\,,$$

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

$$\varphi_2(\boldsymbol{v})\varphi_3(\boldsymbol{x}) - \varphi_3(\boldsymbol{v})\varphi_2(\boldsymbol{x}) = 0\,, \quad \boldsymbol{x} \in \mathbb{R}^2\,.$$

If $\boldsymbol{v}$ is contained in δ, then this line intersects the edge connecting $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$, splitting δ into two subtriangles. The points $\boldsymbol{x} \in \delta$ satisfying $\varphi_2(\boldsymbol{v})\varphi_3(\boldsymbol{x}) \leq \varphi_3(\boldsymbol{v})\varphi_2(\boldsymbol{x})$ fall in the subtriangle that contains $\boldsymbol{v}_2$, while the remaining

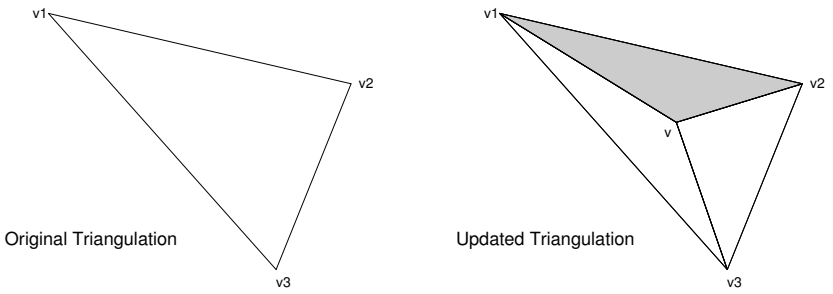

FIGURE 9.13. Adding a new vertex at the point $\boldsymbol{v} = \varphi_1(\boldsymbol{v})\boldsymbol{v}_1 + \varphi_2(\boldsymbol{v})\boldsymbol{v}_2 + \varphi_3(\boldsymbol{v})\boldsymbol{v}_3$. In this case, we are adding to G the continuous, piecewise linear function that takes on the value one at the point $\boldsymbol{v}$ and zero at each of $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$.

points in δ belong to the subtriangle containing $\boldsymbol{v}_3$. Similar statements can be made about lines connecting $\boldsymbol{v}$ and the other vertices $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$.

With this relationship in mind, we define the quantities

$$\varphi_1^*(\boldsymbol{x}) \;=\; \frac{\varphi_1(\boldsymbol{x})}{\varphi_1(\boldsymbol{v})}\,, \quad \varphi_2^*(\boldsymbol{x}) \;=\; \frac{\varphi_2(\boldsymbol{x})}{\varphi_2(\boldsymbol{v})}\,, \quad \text{and} \quad \varphi_3^*(\boldsymbol{x}) \;=\; \frac{\varphi_3(\boldsymbol{x})}{\varphi_3(\boldsymbol{v})}\,.$$

From the discussion in the previous paragraph, we see that the points $\boldsymbol{x} \in \delta$ that fall within the triangular subregion with vertices $\boldsymbol{v}$, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ (the shaded area in Figure 9.13) satisfy the relationship $\varphi_3^*(\boldsymbol{x}) \le \varphi_1^*(\boldsymbol{x})$ and $\varphi_3^*(\boldsymbol{x}) \le \varphi_2^*(\boldsymbol{x})$. Applying (9.3.2) in Section 9.3.1, we also find that within this region, the new basis function $B(\boldsymbol{x})$ is given by $\varphi_3^*(\boldsymbol{x})$. Similar expressions can be derived for the remaining two subtriangles, yielding the following simple rule for constructing $B(\boldsymbol{x})$:

$$B(\boldsymbol{x}) \;=\; \begin{cases} \varphi_3^* & \text{if } \varphi_3^* \le \varphi_1^* \text{ and } \varphi_3^* < \varphi_2^*\,, \\ \varphi_1^* & \text{if } \varphi_1^* \le \varphi_2^* \text{ and } \varphi_1^* < \varphi_3^*\,, \\ \varphi_2^* & \text{if } \varphi_2^* \le \varphi_1^* \text{ and } \varphi_2^* < \varphi_3^*\,. \end{cases}$$

Using these expressions, it is easy to construct $B(\boldsymbol{x})$ from the existing basis elements associated with the vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$. When $\boldsymbol{v}$ is on the boundary of δ, at least one of the barycentric coordinates of $\boldsymbol{v}$ is zero. In this case, one of $\varphi_i^*(\cdot)$ must be infinite and the conditions listed above simplify. For example, if $\boldsymbol{v}$ is on the edge connecting $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, then $\varphi_3^*(\boldsymbol{x})$ is infinite, and we find that within δ,

$$B(\boldsymbol{x}) \;=\; \begin{cases} \varphi_l^* & \text{if } \varphi_1^* \le \varphi_2^*\,, \\ \varphi_2^* & \text{if } \varphi_2^* < \varphi_1^*\,. \end{cases}$$

This set of equations creates $B(\boldsymbol{x})$ for $\boldsymbol{x} \in \delta$. If $\boldsymbol{v}$ is on the boundary of δ, we might also have to produce a similar set of equations to construct

$B(\boldsymbol{x})$ for $\boldsymbol{x}$ belonging to a neighboring triangle of δ. Since 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 formulae for computing the Rao statistic for adding $\boldsymbol{v}$ to the partition $\triangle$.

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 9.13, suppose that we want to add a vertex $\boldsymbol{v}$ on the interior of δ. Now, if we let $B_1(\boldsymbol{x})$, $B_2(\boldsymbol{x})$, and $B_3(\boldsymbol{x})$ represent the piecewise linear basis functions associated with the points $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$ in the updated triangulation, then it is straightforward to demonstrate that, for all points $\boldsymbol{x}$ in the shaded triangle on the right in Figure 9.13,

$$\varphi_1(\boldsymbol{x}) = B_1(\boldsymbol{x}) + \varphi_1(\boldsymbol{v})B_3(\boldsymbol{x})\,, \;\; \varphi_2(\boldsymbol{x}) = B_2(\boldsymbol{x}) + \varphi_2(\boldsymbol{v})B_3(\boldsymbol{x})\,,$$

and

$$\varphi_3(\boldsymbol{x}) = \varphi_3(\boldsymbol{v})B_3(\boldsymbol{x})\,.$$

We have seen the last equation in the definition of the new basis function $B(\boldsymbol{x})$. Similar expressions can be obtained for the remaining two unshaded regions in δ and can be easily extended when $\boldsymbol{v}$ is on a boundary of δ. Again, because so much is known about $\varphi_1(\boldsymbol{x})$, $\varphi_2(\boldsymbol{x})$ and $\varphi_3(\boldsymbol{x})$ from the previous step in the addition process, simple and efficient updating rules can be created for generating the new set of basis functions.

9.3.5 Stepwise deletion

When discussing strategies for reducing the dimension of a space of continuous, piecewise linear splines, we have so far 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 9.10 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 $\triangle$ in an attempt to reduce the dimension of G, there may not be a way to reconnect the remaining vertices to form $\triangle_0$ so that the updated space G_0 is a subspace of G. For example, the central vertex in any of the panels of Figure 9.10 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 hand corner, they can be immediately removed and still produce the proper nesting of spaces. Only vertices falling into one of the three categories listed in

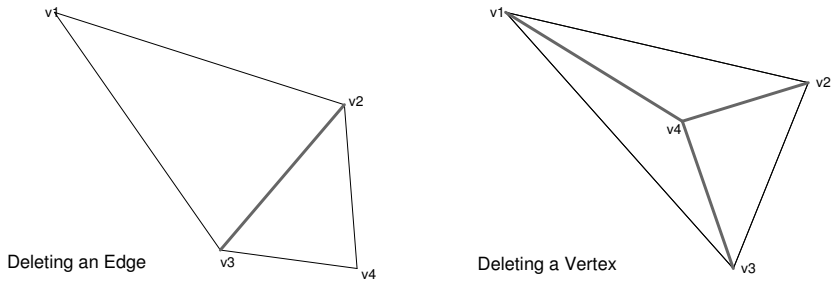

FIGURE 9.14. The effect of enforcing the constraint that functions in G be continuously differentiable across edges in two triangulations.

Figure 9.10 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 9.10, 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 hand corner of the figure. These are the only cases for which this equivalence exists. (The strategy that we employ in the examples in Section 9.4 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 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 $\triangle$ 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 $\triangle$. To make this more precise, consider the triangulation on the left in Figure 9.14 and let $\varphi_1(\boldsymbol{x})$, $\varphi_2(\boldsymbol{x})$, and $\varphi_3(\boldsymbol{x})$ denote the barycentric coordinates of a point $\boldsymbol{x} \in \mathbb{R}^2$ relative to the triangle with vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$. Given a function $g \in G$, let θ_1, θ_2, and θ_3 denote the coefficients of the basis functions associated with these vertices. Then for all points $\boldsymbol{x}$ in this triangle, $g(\boldsymbol{x})$ is the linear function given by

$\theta_1\varphi_1(\boldsymbol{x}) + \theta_2\varphi_2(\boldsymbol{x}) + \theta_3\varphi_3(\boldsymbol{x})$. Now, if we let θ_4 denote the coefficient of the basis function of G associated with the vertex $\boldsymbol{v}_4$, then $g(\boldsymbol{v}_4) = \theta_4$. Therefore, the function g is linear on the union of the two triangles in left hand portion of Figure 9.14 provided that

$$\theta_4 \;=\; g(\boldsymbol{v}_4) \;=\; \theta_1\varphi_1(\boldsymbol{v}_4) + \theta_2\varphi_2(\boldsymbol{v}_4) + \theta_3\varphi_3(\boldsymbol{v}_4)\,.$$

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

$$\theta_1 \;=\; g(\boldsymbol{v}_1) \;=\; \theta_2\tilde{\varphi}_2(\boldsymbol{v}_1) + \theta_3\tilde{\varphi}_3(\boldsymbol{v}_1) + \theta_4\tilde{\varphi}_4(\boldsymbol{v}_1)\,,$$

where $\tilde{\varphi}_2(\boldsymbol{x})$, $\tilde{\varphi}_3(\boldsymbol{x})$, and $\tilde{\varphi}_4(\boldsymbol{x})$ denote the barycentric coordinates of a point $\boldsymbol{x}$ relative to the triangle with vertices $\boldsymbol{v}_2$, $\boldsymbol{v}_3$, and $\boldsymbol{v}_4$. 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 $\triangle$, but we have not produced a simpler triangulation in the process.

Suppose instead that we want to remove the vertex $\boldsymbol{v}_4$ in the middle of the triangle in the right hand portion of Figure 9.14 . Given $g \in G$ and $1 \le i \le 4$, we again let θ_i correspond to the coefficient of the basis function associated with the vertex $\boldsymbol{v}_i$. 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

$$\theta_4 \;=\; \varphi_1(\boldsymbol{v}_4)\theta_1 + \varphi_2(\boldsymbol{v}_4)\theta_2 + \varphi_3(\boldsymbol{v}_4)\theta_3\,, \tag{9.3.11}$$

where $\varphi_1(\boldsymbol{x})$, $\varphi_2(\boldsymbol{x})$ and $\varphi_3(\boldsymbol{x})$ are the barycentric coordinates of a point $\boldsymbol{u}$ relative to the outer triangle in Figure 9.14 . Observe that the expression on the right is the value at $\boldsymbol{v}_4$ of the unique linear function interpolating θ_1, θ_2 and θ_3 at the points $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$, respectively. Recalling that $g(\boldsymbol{v}_4) = \theta_4$, we see that the constraint in (9.3.11) has considerable intuitive appeal.

9.4 The example revisited

The complete experiment involves just 47 data points. Since we have so little data, we do not want to use too many basis functions in our initial model. Therefore, we took as $\triangle_0$ a 15% enlargement of the smallest triangle that contained all 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 11a 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. Since the data set is so small, it seemed reasonable

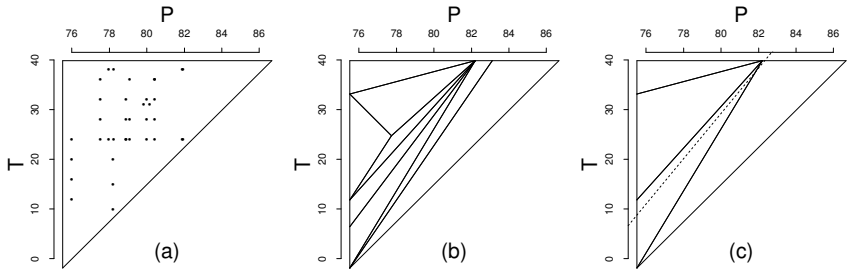

FIGURE 9.15. Initial triangulation (a), largest triangulation (b), and final triangulation (c) for the crystal data. The dashed line in panel (c) is the edge fitted by Cleveland and Fuentes (1996).

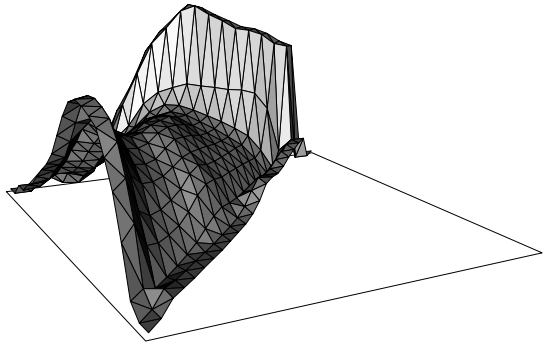

FIGURE 9.16. Rao statistics for the first added vertex (right) for the crystal data.

to consider a somewhat smaller number of possible new vertices than in the simulated example above, and so we set $K = 4$.

The largest model that was fitted had only 9 vertices, since none of the triangles could be further subdivided without violating the requirement on the minimum number of data points. The GCV criterion with penalty parameter 4 selected a Triogram model with six vertices. The largest triangulation encountered during the addition phase and the triangulation associated with the best model are shown in Figures 9.15b and 9.15c, respectively. A perspective plot of the fit is given in Figure 9.4.

The Rao surface introduced in Section 9.3.4 is a useful diagnostic for uncovering structure in this data. In Figure 9.16b we have evaluated the Rao statistic associated with adding a vertex at the points (9.3.10) for $K = 20$ and have connected the points with a continuous, piecewise linear surface. (Recall that in the regression context, the Rao statistic is simply the amount by which the residual sum of squares drops after the addition

of a new basis function.) Notice that the 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 is sparse and the edge in question is a boundary of the initial triangulation. As mentioned earlier, Cleveland and Fuentes (1996) fit two "hinged" planes and thus one interior edge to this data. They find 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 the rightmost panel in Figure 9.15. The Triogram algorithm places an edge in almost the same location, and in fact if we follow the more aggressive deletion scheme outlined above, we can obtain a model very similar to that derived by Cleveland and Fuentes.

9.5 Simulation results

We now present a number of examples to further illustrate the Triogram methodology. 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. Our first example involves data simulated from a bivariate regression model proposed by Gu, Bates, Chen, and Wahba (1990). The design consists of 300 "semi-random" points $\mathbf{x}_i = (x_{1i}, x_{2i})$ in the unit square. At each point $\mathbf{x}_i$ our response is $y_i = f(\mathbf{x}_i) + \epsilon_i$, where the true regression function f is given by

$$\frac{40\exp\{8[(x_1-0.5)^2+(x_2-0.5)^2]\}}{\exp\{8[(x_1-0.2)^2+(x_2-0.7)^2]\}+\exp\{8[(x_1-0.7)^2+(x_2-0.2)^2]\}},$$

and ϵ_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 Gu, Bates, Chen, and Wahba (1990). For our initial triangulation $\triangle_0$, we divide the unit square into four triangles by drawing in both diagonals, yielding an initial model with five degrees of freedom. In Figure, we present both the design points and $\triangle_0$. (In the three panels in Figure 9.17, the point (1,1) corresponds to the bottom left corner of each plot.) Since this data set 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 (9.3.10) and entertain new vertices at the points given in the right hand panel of Figure 9.11. The maximum number of vertices was set

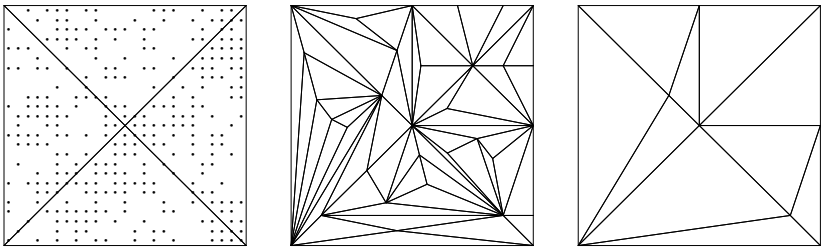

FIGURE 9.17. Initial triangulation (a), largest triangulation (b) and final triangulation (c) for the simulated example.

equal to 35, although this number was rarely reached in our simulations since we required a minimum number of four data points in each triangle. The penalty parameter a in the GCV criterion (9.3.9) was set equal to 4. While this choice seemed to result in the smallest mean integrated squared error across our simulations, taking a in a neighborhood of 4 yielded very similar results.

While the true surface for the artificial regression function from Gu, Bates, Chen, and Wahba (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. In Figure 9.18 we present five triangulations corresponding to a set of continuous, piecewise linear functions. In each example, the functions take on the value zero at all but one vertex. The values at these remaining vertices are given explicitly in Figure fig:simple. We evaluated each surface at 50, 200 and 1000 randomly sampled points inside the triangle and added standard normal errors to the regression surface. Both the height of the examples in Figure fig:simple 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 data sets on which we can compare the performance of Triograms to other popular surface fitting routines.

While each function in Figure 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 4 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 7 data points in each

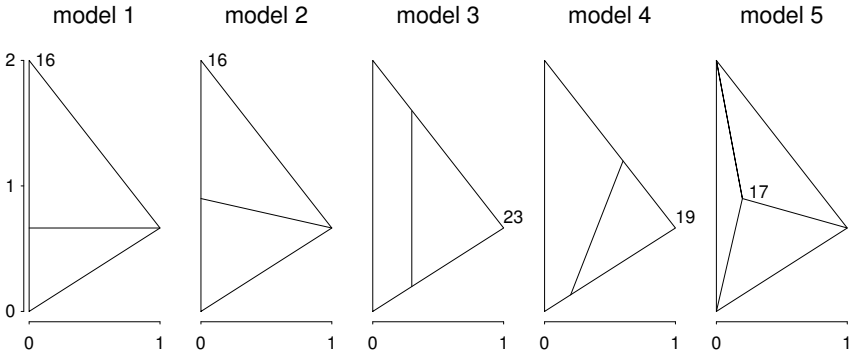

FIGURE 9.18. Five true regression models for a simulation study.

triangle; and for $n = 1000$ we fitted models with at most 20 vertices and at least 10 data points in each triangle.

We computed the mean integrated squared error (MISE) over the 25 simulations for fits from Triogram, MARS (Friedman 1991) and Pimple (Breiman 1991), and the results are summarized in Table 9.1. The typical standard errors of the estimates in Table 9.1 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, while for model 1 MARS wins for $n = 50$ and Triogram wins for $n = 1000$. We should keep in mind that for models 1 and 3 MARS can pick the "correct" model in one step, while several steps would be required for Triogram, since 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 $n = 1000$. In this context, Triogram models are clearly more natural than 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 upon 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, Bates, Chen, and Wahba (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.

	n	model 1	model2	model 3	model 4	model5
Triogram	50	0.1649	0.1706	0.6938	0.2707	0.6662
Pimple[1]	50	0.2347	0.4101	0.2348	0.8172	3.0816
MARS[2]	50	0.1098	0.4192	0.2439	1.0294	2.7530
Triogram	200	0.0447	0.0639	0.1673	0.0232	0.0709
Pimple	200	0.0654	0.1457	0.0805	0.1877	0.6124
MARS	200	0.0436	0.1363	0.0665	0.2658	0.7242
Triogram	500	0.0081	0.0112	0.0299	0.0090	0.0227
Pimple	500	0.0269	0.0359	0.0336	0.0588	0.2587
MARS	500	0.0103	0.0383	0.0066	0.0806	0.3207

TABLE 9.1. Mean integrated squared error (25 simulations). (Excluded one simulation for model 4 with MISE of 11.0 and one for model 5 with MISE of 36.6. Also, excluded one simulation for model 5 with MISE of 43.6.)

9.6 Extensions

Consider density estimation. The likelihood equations are given in Chapter 6. Here, we extend the framework to allow for each model to have a common base triangulation from which we delete vertices. Talk about the protein data and estimating log-odds ratios.

Density estimation

Let f represent the joint density of $\boldsymbol{X} \in \mathcal{X}$. In this context, the vector $\boldsymbol{W}$ equals $\boldsymbol{X}$, since we do not have a response. Now, given coefficients $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_J) \in \mathbb{R}^J$, we can define a density $f(\boldsymbol{x}; \boldsymbol{\theta})$ over $\mathcal{X}$ having the form

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \exp\Big(\theta_1 B_1(\boldsymbol{x}) + \cdots + \theta_J B_J(\boldsymbol{x}) - C(\boldsymbol{\theta}) \Big),$$

where

$$C(\boldsymbol{\theta}) = \int_{\mathcal{X}} \exp\Big(\theta_1 B_1(\boldsymbol{x}) + \cdots + \theta_J B_J(\boldsymbol{x}) \Big) d\boldsymbol{x}$$

is the normalizing constant. Therefore, based on a sample $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ from the distribution of $\boldsymbol{X}$, we estimate f by the function $\widehat{f} = f(\cdot; \widehat{\boldsymbol{\theta}})$, where $\widehat{\boldsymbol{\theta}}$ is chosen to maximize the log-likelihood

$$l_n(\widehat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \log f(\boldsymbol{X}_i; \boldsymbol{\theta}) \,.$$

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

$$E_{\widehat{\boldsymbol{\theta}}} B_j(\boldsymbol{X}) = E_n B_j(\boldsymbol{X}), \qquad 1 \le j \le J, \tag{9.6.1}$$

where

$$E_{\widehat{\boldsymbol{\theta}}} B_j(\boldsymbol{X}) \;=\; \int_{\mathcal{X}} B_j(\boldsymbol{x}) f(\boldsymbol{x};\boldsymbol{\theta}) d\boldsymbol{x} \quad \text{and} \quad E_n B_j(\boldsymbol{X}) \;=\; \frac{1}{n}\sum_{i=1}^{n} B_j(\boldsymbol{X}_i)\,.$$

Since the functions B_j are piecewise linear over $\mathcal{X}$, 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 (9.6.1). To obtain the Hessian associated with this problem, we have to compute quantities of the form $E_{\widehat{\boldsymbol{\theta}}}[B_{j_1}(\boldsymbol{X})B_{j_2}(\boldsymbol{X})]$ for $1 \le j_1, j_2 \le J$, which again have closed form expressions because our basis functions are piecewise linear.

In the top panel of Figure9.6, we present three data sets that are natural candidates for Triogram density estimation. The points in these plots represent a collection of amino acids obtained form 100 protein structures taken from the Brookhaven Protein Data Bank (Hobohm, Scharf, Schneider, and Sander 1992). In order 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 (whether the protein is twisting around a helix, for example), the fraction of the amino acid side-chain area that is buried in the protein structure, and the fraction of the side-chain area that is covered by polar atoms. Since 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 9.6 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. It would seem reasonable, therefore, to attempt to infer the protein's structure from its amino acid sequence. Unfortunately, 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 9.6.

Along the top row of Figure 9.6 we have three data clouds, one corresponding to each local context. There are 591 points in the first plot, 341 in the second and 593 in the third. We first applied the Triogram procedure separately to each dataset corresponding to the three different local envi-

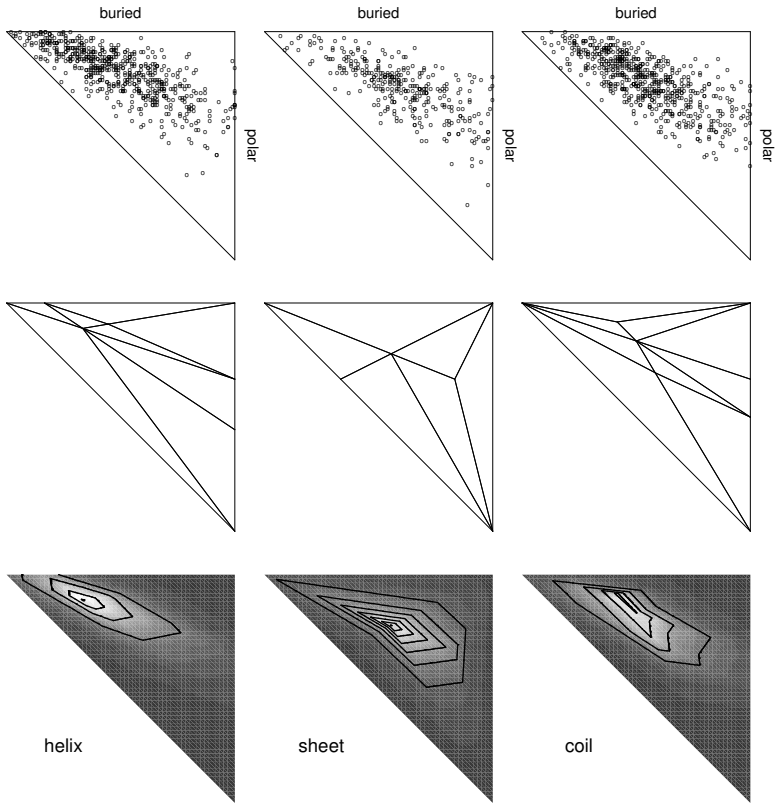

FIGURE 9.19. Triogram density estimates. Separate density estimates are fit for the three local protein contexts. (There were 591 amino acids found in a helix, 341 in sheets and 593 in coils.)

ronments. At each step in the addition process, the set of candidate vertices consisted of the points with barycentric coordinates given in (9.3.10) with $K = 5$ relative to each of the triangles in the current triangulation $\triangle$. 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 (9.3.8). In each case, the best fits were encountered during the stepwise deletion. The underlying triangulations for these final models are plotted in the second row of Figure 9.6, with contour plots of the corresponding densities given in the last row of the same figure. While the piecewise linear character of our Triogram models makes these plots somewhat jagged, they are clearly capturing the essential features of the data.

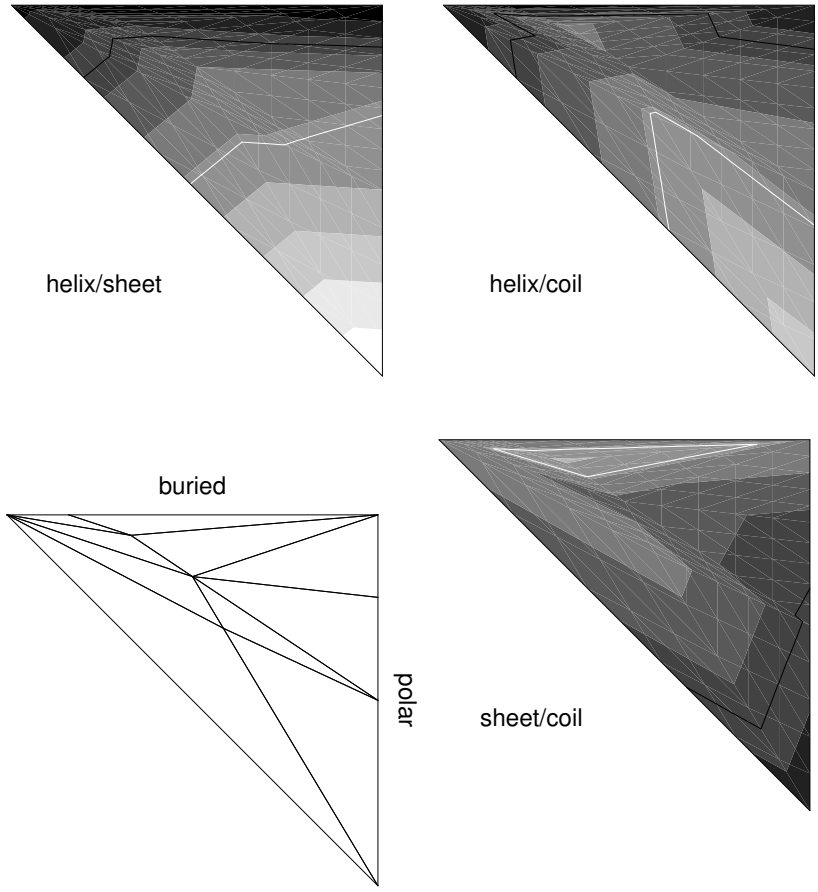

FIGURE 9.20. Log-odds ratios for lysine in the three contexts helix, sheet and coil. In each case, the dark solid lines follow contours with value log(0.5) and the light solid lines follow contours with value log(2).

As mentioned above, 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 data sets simultaneously. For a given triangulation $\triangle$, let $\mathcal{G}$ denote the associated space of continuous, piecewise linear functions. Next, let $\boldsymbol{X}_{ic}$, $i = 1, \ldots, n_c$, denote the observations associated with local environment $c \in \{\text{helix}, \text{sheet}, \text{coil}\}$, and let $\ell_c(\theta_c)$ denote the log-likelihood of these observations as a function of the coefficients θ_c corresponding to the Triogram basis constructed on $\triangle$. During the stepwise addition phase of our model building, we now compute Rao statistics using the likelihood

$$\ell(\,\boldsymbol{\theta}_{\text{helix}}, \boldsymbol{\theta}_{\text{sheet}}, \boldsymbol{\theta}_{\text{coil}}\,) = \ell_{\text{helix}}(\,\boldsymbol{\theta}_{\text{helix}}\,) + \ell_{\text{sheet}}(\,\boldsymbol{\theta}_{\text{sheet}}\,) + \ell_{\text{coil}}(\,\boldsymbol{\theta}_{\text{coil}}\,), \quad (9.6.2)$$

and add the vertex that maximizes this combined Rao statistic. Restrictions on the shape of the resulting triangulations as well as minimum data requirements can be enforced in the obvious way. Our deletion phase again makes use of the log-likelihood in (9.6.2), at each stage deleting the vertex that creates in the smallest increase in the combined Wald statistic. 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 $\triangle$ from which to start the deletion phase.

In Figure 9.20, we present 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 $-2 \approx \log 0.13$ to $3 \approx \log 20$. The dark and light lines intersect the surfaces at $\log 0.5$ and $\log 2$, respectively. 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 $\log 0.2$, since the left top corner of this panel is very dark gray.

Now, consider the difference between Lysine found in a helix and Lysine occurring in a sheet. While the scatterplots in Figure 9.6 indicates 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 9.20 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, since 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.