DEPARTMENT OF STATISTICS

University of Wisconsin
1210 W. Dayton St.
Madison, WI 53706

TECHNICAL REPORT NO. 756

February 1985

INEQUALITY-CONSTRAINED MULTIVARIATE SMOOTHING SPLINES WITH APPLICATION TO THE ESTIMATION OF POSTERIOR PROBABILITIES

by

Miguel Villalobos and Grace Wahba

This research supported by the Office of Naval Research under Grant No. N00014-77-C-0675 and by NASA under Grant No. NAG5-316.

Inequality-Constrained Multivariate Smoothing Splines with Application to the Estimation of Posterior Probabilities

Miguel A. Villalobos
Scientific Center
IBM Mexico

Grace Wahba
Department of Statistics
University of Wisconsin

ABSTRACT

Let $z_i = f(y_1(i),y_2(i)) + \varepsilon_i$, $i = 1,2,\ldots,n$, where f is known to be a "smooth" function of (y_1,y_2) and the ε_i are independent, zero mean random variables. In addition f is known to satisfy a family of linear inequality constraints, for example $0 \leq f(y_1,y_2) \leq 1$, $(y_1,y_2)\varepsilon\Omega \subset E^2$. We propose that f be estimated as the minimizer of

$$\frac{1}{n}\sum_{i=1}^{n} w_i(z_i - f(y_1(i),y_2(i))^2 + \lambda J_m(f)$$

subject to f satisfying the constraints. J_m is the thin plate penalty functional. A good value of λ is estimated by the method of generalized cross validation (GCV) for constrained problems. A characterization of the solution to the minimization problem with the constraints discretized is obtained from known results. We provide a numerical algorithm for computing the GCV estimate of λ and the solution to the (discretized) minimization problem. The method is applied to the estimation of posterior probabilities in the classification problem. Numerical results for both synthetic and experimental data are given.

Key Words: thin plate splines; inequality constraints; cross validation; constrained surface estimation

This research supported by the Office of Naval Research under Grant No. N00014-77-C-0675 and by NASA under Grant No. NAG5-316.

1. INTRODUCTION

The cross validated multidimensional thin plate smoothing spline (Wahba, 1979, Wahba and Wendelberger, 1980) has turned out to be a useful tool to model a smooth but otherwise unknown function of two or more variables. (See, for example Hutchinson and Bischof (1983)).

In two variables this smoothing thin plate spline is associated with the model

$$z_i = f(y_1(i),y_2(i)) + \varepsilon_i,\ i = 1,2,\ldots,n$$

where f is a "smooth" function in a sense to be defined and the ε_i are independent, zero mean random variables with common unknown variance. The estimate $f_{n\lambda}$ of f is the minimizer, in an appropriate space, of

$$\frac{1}{n}\sum_{i=1}^{n}(z_i - f(y_1(i),y_2(i)))^2 + \lambda J_m(f) \qquad (1.1)$$

where

$$J_m(f) = \int_{-\infty}^{\infty}\int\sum_{i=1}^{n}\binom{m}{\nu}\left(\frac{\partial^m f}{\partial y_1^{\nu}\partial y_2^{m-\nu}}\right)^2 dy_1 dy_2,$$

and a good value of the smoothing parameter λ is obtained by the method of generalized cross validation (GCV). It has occurred to a number of workers that it would be useful to combine spline smoothing (both univariate and multivariate) with the imposition of linear inequality constraints, for example nonnegativity, monotonicity, etc. Although some of the theory for doing this has been available for some time (see Kimeldorf and Wahba 1971), what is needed is an efficient computational algorithm. Wegman and Wright (1983), in the context of isotonic regression, say:

> Computational algorithms are clearly the stumbling block in further development of the theory of isotonic splines. When such algorithms become available we believe that smooth, order-preserving non-parametric estimators will substantially enhance the efficiency of estimation procedures currently in use.

In this paper we demonstrate the feasibility of doing large multidimensional smoothing problems like (1.1) with inequality constraints. The computational algorithm developed here can be used in applications such as survival curve estimation, logistic regression and the estimation of posterior probabilities. We develop here an algorithm for an inequality constrained, cross validated multidimensional smoothing spline and, as an example, we apply it to the estimation of posterior probabilities.

We believe that the methods presented here will be useful for exploring properties of the data and presenting them in a way comprehensible to the layman.

In §2 we will present the general inequality-constrained thin plate spline smoothing problem, and we will describe the method of generalized cross-validation for constrained problems to choose the smoothing parameter.

In §3, we discuss the actual computation of the spline.

In §4 we apply the results to the estimation of posterior probabilities in the classification problem. Two numerical examples, one with synthetic and one with experimental data are presented.

2. INEQUALITY CONSTRAINED THIN PLATE SPLINES

2.1 The General Minimization Problem

For f a function of d variables, the thin plate spline penalty functional $J_m(f)$ is defined as

$$J_m(f) = \sum_{\substack{d \\ \sum_{j=1} \alpha_j = m}} \left[\frac{m!}{\alpha_1! \ldots \alpha_d!}\right] \int \ldots \int \left[\frac{\partial^m f}{\partial y_1^{\alpha_1} \ldots \partial y_d^{\alpha_d}}\right]^2 dy_1, \ldots, dy_d. \qquad (2.1)$$

We say that f is "smooth" if $J_m(f)$ is not too large. More precisely we will be assuming that $f \in H(m,d)$ which is the vector space of all the Schwartz distributions for which all the partial derivatives in the distributional sense of total order m are square integrable [see Meinguet (1979)]. Consider the model

$$z_i = L_i f + \varepsilon_i, \quad i = 1,2,\ldots,n \qquad (2.2)$$

where L_i is a bounded linear functional on H(m,d) and the ε_i are zero mean independent random variables with variance $\sigma^2 \sigma_i^2$, $i = 1,\ldots,n$, where σ^2 is an unknown constant. If $2m-d>0$ then the evaluation functionals $L_i f = f(y(i))$, $y(i) = (y_1(i),\ldots,y_d(i))$, are all bounded, $i = 1,\ldots,n$. More generally if $2m-2k-d>0$ then the mixed partial derivatives of total order k are bounded. (See, e.g. Wahba and Wendelberger (1980)). Suppose it is also known that f is in some closed convex set C that can be well approximated by the set

$$C_k = \{f: N_j f \leq r_j, \ j = 1,\ldots,k\}$$

where $N_1,\ldots,N_k$ are also bounded linear functionals. For example suppose $C = \{f: 0 \leq f(y), \ y \in \Omega\}$ where Ω is some interval in E^d. Then C may be approximated

by $C_k = \{f: 0 \leq f(s(j)),\ j = 1,2,\ldots,k\}$ where the $\{s(j)\}$ form a fine regular grid in Ω. See Wahba (1973) for a partial result concerning the goodness of this approximation.

Given the data $\{z_i\}$ and the set C_k we will estimate f by $f_{n\lambda}$, the solution to problem 2.1:

Problem 2.1: Find $f \in H(m,d)$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n}(z_i - L_i f)^2/\sigma_i^2 + \lambda J_m(f)$$

subject to $N_j f \leq r_j$ $j = 1,\ldots,k$.

The solution to problem 2.1 is expressible in terms of polynomials of total degree less than m, and the fundamental solutions of the iterated Laplacian. We call the function $f_{n\lambda}$ that solves problem 2.1 a "constrained thin plate smoothing spline". Before stating the result we introduce some notation.

Let $H_o(m,d)$ be the space of polynomials on R^d of total degree less than m. Then $H_o(m,d)$ is an M-dimensional space, where $M = \binom{m+d-1}{d}$.

Let $\phi_1,\ldots,\phi_M$ be the M monomials of degree less than m given by

$$\phi_\ell(t) = t_1^{\alpha_1}\ldots t_d^{\alpha_d}$$

$$t = (t_1,\ldots,t_d)$$

$$\sum_{j=1}^{d}\alpha_j < m.$$

Observe that $J_m(\phi_\ell) = 0$, $\ell = 1,\ldots,M$, so that polynomials of total degree less than m are considered infinitely smooth by the penalty functional J_m.

Let the Laplacian Δ be

$$\Delta f = \sum_{i=1}^{d} \frac{\partial^2 f}{\partial y_i^2} \quad .$$

If f and all its derivatives up to order m-1 are continuous and are zero at infinity, then, by integration by parts, one has

$$J_m(f) = \int \ldots \int f \Delta^m f.$$

If $s,t \in R^d$, the fundamental solution of the iterated Laplacian is given by $E_m(s,t)$ defined by

$$E_m(s,t) = \begin{cases} \theta_m ||s-t||^{2m-d} \ln(||s-t||) & \text{if d is even} \\ \theta_m ||s-t||^{2m-d} & \text{if d is odd} \end{cases} \tag{2.3}$$

where

$$\theta_m = \begin{cases} \dfrac{(-1)^{d/2+1+m}}{2^{2m-1}\pi^{d/2}(m-1)!(m-d/2)!}, & \text{d even} \\ \dfrac{\Gamma(d/2-m)}{2^{2m}\pi^{d/2}(m-1)!}, & \text{d odd.} \end{cases}$$

We will assume that the following two conditions hold:

Condition 2.1:

$L_1,\ldots,L_n$, $N_1,\ldots,N_k$ are linearly independent continuous linear functionals.

Condition 2.2:

The rank of the matrix $T_1 := (L_i \phi_j)$ $i = 1,\ldots,n$; $j = 1,\ldots,M$ is M.

For example, in the case where $L_i f = f(y(i))$, $i = 1,\ldots,n$ and $N_j f = f(s(j))$, $j = 1,\ldots,k$; if the points $y(1),\ldots,y(n)$, $s(1),\ldots,s(k)$ are all distinct, condition 2.1 will be satisfied. To satisfy condition 2.2 we need that $n \geq M$ and that there is an M-element subset $y(i_1),\ldots,y(i_M)$ of $y(1),\ldots,y(n)$ such that there is exactly one polynomial in $H_o(m,d)$ interpolating to data given at $y(i_1),\ldots,y(i_M)$.

The following theorem gives a representation for the solution of problem 2.1. The proof is a straightforward application of the results in Duchon (1976) to the results in Kimeldorf and Wahba (1971) and can be found in Villalobos (1983).

Theorem 2.1. If conditions 2.1 and 2.2 hold, the solution $f_{n\lambda}$ to problem 2.1 is of the form

$$f_{n\lambda}(t) = \sum_{i=1}^{n} c_i L_{i(s)} E_m(s,t) + \sum_{j=1}^{k} b_j N_{j(s)} E_m(s,t) + \sum_{\ell=1}^{M} d_\ell \phi_\ell(t) \qquad (2.4)$$

Here the subscript "(s)" indicates that the functional L_i (or N_j) is to be applied to what follows considered as a function of s.

Let $D_\sigma^{-2} := \text{diag}\,(1/\sigma_1^2,\ldots,1/\sigma_n^2)$ and let $a = (c_1,\ldots,c_n, b_1,\ldots,b_k)^t$, then it can be shown that the coefficients $c_1,\ldots,c_n$, $b_1,\ldots,b_k$ and $d_1,\ldots,d_M$ in Theorem 2.1 are obtained by solving the following quadratic programming:

Problem 2.2.

Minimize $G(a,d)$ subject to $g(a,d) \leq 0$ and $T^t a = 0$, where now,

$$G(a,d) = \frac{1}{2}(E_1 a + T_1 d - z)^t D_\sigma^{-2}(E_1 a + T_1 d - z) + \frac{n\lambda}{2} a^t E a$$

$$g(a,d) = E_2 a + T_2 d - r$$

where:

$$E = \begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix}, \qquad T = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix},$$

$E_1 = [E_{11} : E_{12}]$, $E_2 = [E_{21} : E_{22}]$ and $E_{21} = E_{12}^t$. E_{11}, E_{12}, E_{22}, T_1 and T_2 are given in Table 2.1, and $r = (r_1,\ldots,r_k)^t$.

TABLE 2.1

Matrix	Dimension	(i,j)th element	
E_{11}	n×n	$L_{i(s)} L_{j(t)} E_m(s,t)$	i = 1,...,n j = 1,...,n
E_{12}	n×k	$L_{i(s)} N_{j(t)} E_m(s,t)$	i = 1,...,n j = 1,...,k
E_{22}	k×k	$N_{i(s)} N_{j(t)} E_m(s,t)$	i = 1,...,k j = 1,...,k
T_1	n×M	$L_i \phi_j$	i = 1,...,n j = 1,...,M
T_2	k×M	$N_i \phi_j$	i = 1,...,k j = 1,...,M

For a given value of λ we can solve problem 2.2 using a high quality quadratic programming routine.

2.2 The Choice of the Smoothing Parameter

In real life problems the correct value of the smoothing parameter λ is not known. Wahba and Wold (1975), Craven and Wahba (1979) and Golub, Heath and Wahba

(1979) have suggested the use of generalized cross-validation to estimate λ from the data in the unconstrained case. In the presence of linear constraints, Wahba (1980) and (1982) suggested the use of generalized cross-validation for constrained problems.

To aid in the description of the method of generalized cross-validation for constrained problems, which we will refer to as GCVC, we first give a brief review of the method of generalized cross-validation for unconstrained problems which will be referred to as GCV.

Let $f_{n\lambda}^{[q]}$ be the minimizer of

$$\frac{1}{n} \sum_{i=1, i\neq q}^{n} (L_i f - z_i)^2 + \lambda J_m(f).$$

If λ is a good choice, then, on the average, $(L_q f_{n\lambda}^{[q]} z_q)^2$ should be small and so the ordinary cross-validation function $V_o(\lambda)$ defined by

$$V_o(\lambda) = \frac{1}{n} \sum_{q=1}^{n} (L_q f_{n\lambda}^{[q]} - z_q)^2 \tag{2.5}$$

should be small.

Craven and Wahba (1979) and Golub, Heath and Wahba (1979) showed that

$$V_o(\lambda) \equiv \frac{1}{n} \sum_{i=1}^{n} \frac{(L_i f_{n\lambda} - z_i)^2}{(1-a_{ii}(\lambda))^2} \tag{2.6}$$

where $f_{n\lambda}$ is the minimizer of

$$Q_\lambda(f) = \frac{1}{n} \sum_{i=1}^{n} (L_i f - z_i)^2 + \lambda J_m(f)$$

and $a_{ii}(\lambda)$ is the (i,i) entry of the n×n "influence matrix" $A(\lambda)$ satisfying

$$\begin{pmatrix} L_1 f_{n\lambda} \\ \vdots \\ L_n f_{n\lambda} \end{pmatrix} = A(\lambda)z$$

Craven and Wahba (1979) and Golub, Heath and Wahba (1979) show that from the point of view of minimizing the predictive mean square error given by

$$T(\lambda) = \frac{1}{n} \sum_{i=1}^{n} (L_i f_{n\lambda} - L_i f)^2,$$

$V_0(\lambda)$ should be replaced by the generalized cross-validation function $V(\lambda)$ given by:

$$V(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \frac{(L_i f_{n\lambda} - z_i)^2}{(1-a_{ii}(\lambda))^2} \omega_i^2(\lambda) = \frac{\frac{1}{n}||(I-A)(\lambda))z||^2}{[\frac{1}{n}tr(I-A(\lambda))]^2}$$

where

$$\omega_i(\lambda) = \frac{1-a_{ii}(\lambda)}{1-\frac{1}{n}\sum_{j=1}^{n} a_{jj}(\lambda)}$$

They showed that the minimizer of (2.6) estimates the minimizer of $T(\lambda)$. Since then a number of authors have examined properties of the minimizer of $V(\lambda)$, see the references in Wahba (1983). In addition $V(\lambda)$ is in general substantially faster to compute than $V_0(\lambda)$.

Now let C be any closed and convex set in H(m,d), $f_{n\lambda}$ be the minimizer of $Q_\lambda(f)$ in C and let $f_{n\lambda}^{[q]}$ be the minimizer in C of

$$\frac{1}{n} \sum_{i=1;i\neq q} (L_i f - z_i)^2 + \lambda J_m(f). \tag{2.7}$$

The ordinary cross-validation function is given by $V_o(\lambda) = \frac{1}{n} \sum_{q=1}^{n} (L_q f_{n\lambda}^{[q]} - z_q)^2$. It is obvious that $V_o(\lambda)$ would be prohibitive to compute in most cases.

It can be shown by following the proof of lemma 3.1 in Craven and Wahba (1979) that given the data

$$[z_1,\ldots,z_{q-1},L_q f_{n\lambda}, z_{q+1},\ldots,z_n]^t$$

the minimizer of $Q_\lambda(f)$ in C is $f_{n\lambda}^{[q]}$, that is,

$$f_{n\lambda}[z+\delta_q] = f_{n\lambda}[z]. \tag{2.8}$$

The notation $f_{n\lambda}[z+\delta_q]$ indicates that $f_{n\lambda}$ is the minimizer in C of $Q_\lambda(f)$ based on the data vector $z+\delta_q$, where δ_q is given by:

$$\delta_q = (0,\ldots,L_q f_{n\lambda}^{[q]}[z]-z_q,\ldots,0)^t,$$

and $f_{n\lambda}^{[q]}[z]$ is the minimizer in C of (2.7) based on the data vector z.

Using (2.8), it is not hard to show that the ordinary cross-validation function V_o in the constrained case can be written as

$$V_o(\lambda) = \frac{1}{n} \sum \frac{(L_q f_{n\lambda} - z_q)^2}{(1-a_{qq}^*(\lambda))^2}$$

where

$$a_{qq}^*(\lambda) = \frac{L_q f_{n\lambda}[z+\delta_q] - L_q f_{n\lambda}[z]}{L_q f_{n\lambda}^{[q]}[z] - z_q}$$

is what Wahba calls the "differential influence" of z_q when λ is used.

The GCVC function is obtained by replacing a^*_{qq} by the average "differential influence", so that the GCVC estimate of λ is obtained by minimizing

$$V^C(\lambda) = \frac{\frac{1}{n}\sum_{i=1}^{n}(L_i f_{n\lambda} - z_i)^2}{(1-\frac{1}{n}\sum_{q=1}^{n} a^*_{ii}(\lambda))^2} .$$

As we mentioned earlier we will assume that the convex set C can be well approximated by the intersection of a finite number of half spaces:

$$C_k = \{f: N_i f \leq r_i,\ i = 1,\ldots,k\}.$$

Then, to evaluate $V^C(\lambda)$ for a single value of λ we need to solve n quadratic programming problems in n+k-M variables. To avoid this, Wahba (1982) suggested using the appoximate generalized cross-validation function given by:

$$V^C_{app}(\lambda) = \frac{\frac{1}{n}\sum_{i=1}^{n}(L_i f_{n\lambda} - z_i)^2}{(1-\frac{1}{n}\sum_{q=1}^{n} a_{qq}(\lambda))^2} \qquad (2.9)$$

where

$$a_{qq}(\lambda) = \frac{\partial}{\partial z_q} L_q f_{n\lambda}\Big|_z .$$

We note that unfortunately $\sum_{q=1}^{n} a_{qq}(\lambda)$ of (2.9) is not necessarily a continuous function of λ. A discontinuity will generally occur as λ changes, when the set of active constraints changes. However, this has not caused

serious problems in the examples tried. Once the set of active constraints for a fixed λ has been determined, it will be seen below that $\sum a_{qq}(\lambda)$ in (2.9) can be found from $A(\lambda)$, the influence matrix for the corresponding minimization problem obtained by setting the active inequality constraints as equality constraints.

3. THE ALGORITHM

3.1 Restatement of the problem.

The software written as part of this work was developed for the case where the functionals $L_1,\ldots,L_n$ and $N_1,\ldots,N_k$ are evaluation functionals. That is, we want to minimize

$$\frac{1}{n}\sum_{i=1}^{n}(f(y(i))-z_i)^2/\sigma_i^2 + \lambda J_m(f)$$

subject to $f(s(i)) \leq r_i$, for $i = 1,\ldots,k$. However, we present the algorithm in its more general form, where the L_i and N_j are any continuous linear functionals.

Before solving problem 2.1 we solve the unconstrained problem, estimating the value of λ by generalized cross-validation. The software to solve the unconstrained problem in the case where the L_i's are evaluation functionals, was developed originally by Wendelberger (1981) and can be obtained from the Madison Academic Computing Center (1981). M.F. Hutchinson (1984) has also developed transportable software, especially suitable for larger data sets, and further numerical methods have been proposed by Bates and Wahba (1982, 1983).

If the solution for the unconstrained problem satisfies all the contraints, then that is also the solution to problem 2.1.

If this is not the case, we use the value of λ obtained from the solution of the unconstrained problem, say $\hat{\lambda}_o$ as a starting guess for the "correct" λ for problem 2.1. In fact, since the imposition of constraints is in some sense a kind of smoothing, it is natural to expect that an optimal λ for the constrained problem will be smaller than $\hat{\lambda}_o$. Also, intuitively, the optimal λ for the constrained problem, say $\hat{\lambda}$ should not be "too far away" from $\hat{\lambda}_o$.

There are two important parts in the algorithm to compute the solution to problem 2.1. One is the solution of a quadratic programming problem of size n+k-M for each value of λ that we consider, and the other is the comptuation of $V^C_{app}(\lambda)$ given by (2.9). These two parts are the most intensive in terms of computational effort and hence it is important to try to make them as efficient as possible.

For computational convenience suppose that instead of observing z, we observe the vector

$$z_\sigma = [z_1/\sigma_1,\ldots,z_n/\sigma_n]^t$$

and define the matrices:

$$T_1^\sigma = D_\sigma^{-1}T_1,\ T^\sigma = \begin{pmatrix} T_1^\sigma \\ T_2 \end{pmatrix},$$

$$E_{11}^\sigma = D_\sigma^{-1}E_{11}D_\sigma^{-1},\ E_{12} = D_\sigma^{-1}E_{12},$$

$$E_1^\sigma = [E_{11}^\sigma:\ E_{12}^\sigma],\ E_2^\sigma = [E_{21}^\sigma:\ E_{22}^\sigma] \text{ and}$$

$$E^\sigma = \begin{pmatrix} E_1^\sigma \\ E_2^\sigma \end{pmatrix}$$

where E_{11}, E_{12}, E_{22}, T_1 and T_2 are given in Table 2.1. After some algebra it is easy to see that problem 2.2 is equivalent to

Problem 3.1: Minimize

$$\tfrac{1}{2}[E_1^\sigma a_\sigma + T_1^\sigma d - z_\sigma]^t[E_1^\sigma a_\sigma + T_1^\sigma d - z_\sigma] + n\lambda a_\sigma^t E^\sigma a_\sigma$$

subject to

$$E_2^\sigma a_\sigma + T_2 d - r \leq 0 \text{ and } T^{\sigma^t} a_\sigma = 0$$

where $a_\sigma = D_\sigma a$.

Let $Q = [Q_1^t : Q_2^t]^t$ and R be the Q-R decomposition of T^σ, that is,

$$\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} T^\sigma = \begin{pmatrix} R \\ 0_{(n+k-M)\times M} \end{pmatrix}.$$

Let e_σ be the n+k-M dimensional vector such that $a_\sigma = Q_2 e_\sigma$. Since $T^{\sigma^t} a_\sigma = 0$, such a representation always exists. Then, instead of solving problem 3.1 we solve the equivalent

Problem 3.2: Minimize

$$G^\sigma(e_\sigma, d) = \tfrac{1}{2}[E_1^\sigma Q_2 e_\sigma + T_1^\sigma d - z_\sigma]^t[E_1^\sigma Q_2 e_\sigma + T_1^\sigma d - z_\sigma] + n\lambda e_\sigma^t Q_2^t E^\sigma Q_2 e_\sigma .$$

subject to

$$g^\sigma(e_\sigma, d) = E_2^\sigma Q_2 e_\sigma + T_2 d \leq r.$$

Then if $(\hat{e}_\sigma, \hat{d})$ solves problem 3.2, $(\hat{a}_\sigma, \hat{d})$ solves problem 3.1, where

$\hat{a}_\sigma^t = [\hat{e}_\sigma^t Q_2^t : \hat{b}^t]$ and then $\begin{pmatrix} \hat{a} \\ \hat{d} \end{pmatrix} = \begin{pmatrix} \hat{c} \\ \hat{b} \\ \hat{d} \end{pmatrix} = \begin{pmatrix} D_\sigma^{-1} c_\sigma \\ \hat{b} \\ \hat{d} \end{pmatrix}$ solves problem 2.2. Now

$$G^\sigma(e_\sigma, d) = \tfrac{1}{2}[e_\sigma : d^t]\, \Gamma \begin{pmatrix} e_\sigma \\ d \end{pmatrix} - z_\sigma [E_1^\sigma Q_2 : T_1] \begin{pmatrix} e_\sigma \\ d \end{pmatrix}$$

where Γ is the Hessian and is given by

$$\begin{pmatrix} Q_2^t E_1^{\sigma t} E_1^\sigma Q_2 + n\lambda Q_2^t E^\sigma Q_2 & Q_2 E_1^{\sigma t} T_1^\sigma \\ T_1^{\sigma t} E_{11}^\sigma Q_2 & T_1^{\sigma t} T_1^\sigma \end{pmatrix}.$$

Finally, we write $g^\sigma(e_\sigma, d)$ as

$$g^\sigma(e_\sigma, d) = [E_2^\sigma Q_2 : T_2] \begin{pmatrix} e_\sigma \\ d \end{pmatrix}.$$

3.2 The Quadratic Programming Algorithm

Let $f_{n\lambda}$, given by

$$f_{n\lambda}(t) = \sum_{i=1}^{n} c_i L_{i(s)} E_m(s,t) + \sum_{j=1}^{k} b_j N_{j(s)} E_m(s,t) + \sum_{\ell=1}^{M} d_\ell \phi_\ell(t) \qquad (3.1)$$

be the solution to problem 2.1 for a given value of λ. Suppose that there are ℓ active constraints at the solution that correspond to $N_{\nu(1)}, \ldots, N_{\nu(\ell)}$, where

$$\gamma := \{\nu(1), \ldots, \nu(\ell)\}$$

is the set of indices of the active constraints.

If we solve problem 2.1 for some other value of λ, say λ', then we will get a possibly different set $\gamma' := \{\nu'(1),\ldots,\nu'(\ell)\}$ corresponding to the active constraints at the solution. If λ and λ' are relatively close, it is likely that the sets γ and γ' will either be the same or at least they will not be "too different".

It is this feature of our problem that motivated the use of an "active set" algorithm to solve the quadratic programming problem. The algorithm that we used was developed by Gill, Gould, Murray, Saunders and Wright (1982, 1984). The idea behind this algorithm is as follows.

If the correct active set of constraints were known a priori then the solution to problem 2.1 would be the solution to a problem with equality constraints. There are several efficient algorithms for solving problems with equality constraints and, in fact, the presence of equality constraints actually reduces the dimensionality in which the optimization occurs. Therefore it is desirable to apply techniques from the equality constrained case to solve problem 2.1. To do this a subset of the constraints of the original problem, called a "working set" of constraints is selected to be treated as equality constraints. Obviously, the ieal candidate for the working set would be the correct active set. Since the correct active set is not available, the method includes procedures for testing whether the current working set is the correct one and altering it, adding or deleting constraints if not.

In our problem, every time we solve the quadratic programming problem for a given value of λ, say λ', we obtain a correct active set for that particular λ. By the argument at the beginning of this section this correct active set will be a good starting guess for the correct active set for some other value of λ close to λ' and therefore once we solve the problem for the first time we can expect

very fast convergence with this active set algorithm. In fact this was what was observed in our Monte Carlo studies. In the cases where the active set did not change from one value of λ to the following, the quadratic programming routine converged in one iteration. We will discuss this further in Section 4 where we will present the results of the Monte Carlo study.

The Fortran routines to solve the quadratic programming problem were kindly provided by Nicholas I.M. Gould and are based on method 3 of Gill, Gould, Murray, Saunders and Wright (1982).

3.3 The Computation of the Approximate GCVC.

Let $f_{n\lambda}$ given by (3.1) be the solution to problem 2.1. Let $N_{\nu(1)},\ldots,N_{\nu(\ell)}$ correspond to the ℓ active constraints at the solution, then $f_{n\lambda}$ is also the solution to

Problem 3.3: Minimize

$$\frac{1}{2}\sum_{i=1}^{n}(L_i f - z_i)^2/\sigma_i^2 + \frac{n\lambda}{2}J_m(f)$$

subject to $N_{\nu(j)} = r_{\nu(j)}$, $j = 1,\ldots,\ell$.

The solution to Problem 3.3 is linear in the components of the data vector z and it is now our purpose to exploit this to obtain an algorithm for computing the term

$$\sum a_{qq}(\lambda) = \sum \frac{\partial}{\partial z_q} L_q f_{n\lambda}$$

which appears in the denominator of V_{app}^{C}. (After the quadratic programming problem has been solved, the numerator of V_{app}^{C} can be computed easily.)

First we must introduce some more notation. Let the matrices $E_{22}(\gamma)$, $E_{12}(\gamma)$ and $T_2(\gamma)$ consist of the rows and columns of the matrices E_{22}, E_{12}^{σ} and T_2, corresponding to the active set of constraints; that is, $E_{22}(\gamma)$ consists of rows and columns $\nu(1),\ldots,\nu(\ell)$ of E_{22}, $E_{12}^{\sigma}(\gamma)$ consists of columns $\nu(1),\ldots,\nu(\ell)$ of E_{12}^{σ} and $T_2(\gamma)$ consists of rows $\nu(1),\ldots,\nu(\ell)$ of T_2. Also define the matrices:

$$T^{\sigma}(\gamma) = \begin{pmatrix} T_1^{\sigma} \\ T_2(\gamma) \end{pmatrix}, \quad E_{21}^{\sigma}(\gamma) = E_{12}^{\sigma}(\gamma)^t,$$

$$E_1^{\sigma}(\gamma) = [E_{11}^{\sigma} : E_{12}^{\sigma}(\gamma)], \quad E_2^{\sigma}(\gamma) = [E_{21}^{\sigma}(\gamma) : E_{22}(\gamma)]$$

and

$$E^{\sigma}(\gamma) = \begin{pmatrix} E_{11}^{\sigma} & E_{12}^{\sigma}(\gamma) \\ E_{21}^{\sigma}(\gamma) & E_{22}(\gamma) \end{pmatrix},$$

where E_{11}^{σ} and T_1^{σ} are as defined in Subsection 3.. Here we use the notation (γ) to emphasize the dependence on the set γ of active constraints.

Now we can rewrite problem 3.1 as:

Problem 3.4

Minimize

$$\overline{G}(a_{\sigma},d) = \tfrac{1}{2}(E_1^{\sigma}(\gamma)a_{\sigma} + T_1^{\sigma}d - z_{\sigma})^t(E_1^{\sigma}(\gamma)a + T_1^{\sigma} - z_{\sigma})$$

$$+ n\lambda a_{\sigma}^t E^{\sigma}(\gamma)a_{\sigma}$$

subject to

$$g(a_\sigma, d) = E_2(\gamma)a_\sigma + T_2(\gamma) - r(\gamma) = 0$$

where $r(\gamma) = (r_{\nu(1)}, \ldots, r_{\nu(\ell)})^t$.

Let $Q_1(\gamma)$, $Q_2(\gamma)$ and $R(\gamma)$ form the Q-R decompositon of $T^\sigma(\gamma)$, that is, they satisfy

$$[Q_1(\gamma)^t : Q_2(\gamma)^t]^t T^\sigma(\gamma) = [R(\gamma)^t : 0^t]^t \tag{3.3}$$

where $Q_1(\gamma)$ is $M \times n+\ell$, $Q_2(\gamma)$ is $n + \ell - M \times n + \ell$, $R(\gamma)$ is $M \times M$ and 0 is an $(n+\ell-M) \times M$ zero matrix. We assume below that $Q_2(\gamma)^t E^\sigma(\gamma) Q_2(\gamma)$ is positive definite. This will hold if conditions 2.1 and 2.2 hold (see Dyn and Wahba (1982)).

Using standard methods (see e.g. Wahba (1978,1980)) it can be shown that a_σ and d satisfy the following system of equations

$$(E^\sigma(\gamma) + n\lambda W)a_\sigma + T^\sigma(\gamma)d = \begin{pmatrix} z \\ r(\gamma) \end{pmatrix} \tag{3.4a}$$

$$T^\sigma(\gamma)^t a_\sigma = 0 \tag{3.4b}$$

where

$$W = \begin{pmatrix} I_n & 0_{n\times e} \\ 0_{e\times n} & 0_{e\times e} \end{pmatrix}.$$

Since a_σ must be in the column space of $Q_2(\gamma)^t$ by (3.4b) we can write $a_\sigma = Q_2(\gamma)^t e_\sigma$ for some $n + \ell - M$ column vector e_σ, and (3.4a) can be rewritten

$$Q_2(\gamma)(E^\sigma(\gamma)+n\lambda W)Q_2(\gamma)^t e_\sigma = Q_2(\gamma)\begin{pmatrix} z \\ r(\gamma) \end{pmatrix}$$

and

$$a_\sigma = Q_2(\gamma)^t e_\sigma = Q_2(\gamma)^t[Q_2(\gamma)(E^\sigma(\gamma)+n\lambda W)Q_2(\gamma)^t]^{-1}Q_2(\gamma)\begin{pmatrix} z \\ r(\gamma)\end{pmatrix}.$$

Now, by substitution in (3.1) we have

$$\begin{pmatrix} \hat{z} \\ r(\gamma)\end{pmatrix} = \begin{pmatrix} L_1 f_{n\lambda} \\ \vdots \\ L_n f_{n\lambda} \\ r(\gamma)\end{pmatrix} = E^\sigma(\gamma)a_\sigma + T^\sigma(\gamma)d$$

and (3.4) gives

$$\begin{pmatrix} z \\ r(\gamma)\end{pmatrix} = (E^\sigma(\gamma) + n\lambda W)a_\sigma + T^\sigma(\gamma)d$$

so

$$\begin{pmatrix} z-\hat{z} \\ 0\end{pmatrix} = n\lambda W a_\sigma = n\lambda W Q_2(\gamma)^t[Q_2(\gamma)(E^\sigma(\gamma) + n\lambda W)Q_2(\gamma)^t]^{-1}Q_2(\gamma)\begin{pmatrix} z \\ r(\gamma)\end{pmatrix}.$$

Let

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22}\end{pmatrix} = Q_2(\gamma)^t[Q_2(\gamma)(E^\sigma(\gamma) + n\lambda W)Q_2(\gamma)^t]^{-1}Q_2(\gamma)$$

where B_{11} is $n\times n$ and B_{22} is $\ell\times\ell$. Then

$$z - \hat{z} = n\lambda(B_{11}z + B_{12}r(\gamma))$$

giving

$$\sum_{q=1}^{n} a_{qq}(\lambda) = Tr(I_n - n\lambda B_{11})$$

or

$$n - \sum_{q=1}^{n} a_{qq}(\lambda) = n\lambda Tr B_{11}.$$

Now

$$\begin{aligned} \mathrm{Tr}B_{11} &= \mathrm{Tr}WB \\ &= \mathrm{Tr}\Delta(\Phi+n\lambda\Delta)^{-1} \end{aligned}$$

where Δ and Φ are the $n + \ell - M$ square matrices

$$\Delta = Q_2(\gamma)^t W Q_2(\gamma)$$

$$\Phi = Q_2(\gamma)^t E^{\sigma}(\gamma) Q_2(\gamma)$$

and it follows that

$$\mathrm{Tr}\Delta(\Phi+n\lambda\Delta)^{-1} = \sum_{i=1}^{n+\ell-M} \frac{\rho_i}{1+n\lambda\rho_i} \qquad (3.5)$$

where $\rho_1,\ldots,\rho_{n+\ell-M}$ are the eigenvalues of the real symmetric eigenvalue problem

$$\Delta\xi_i = \rho_i \Phi\xi_i, \quad i = 1,2,\ldots,n+\ell-M.$$

Using (3.5) we need only solve the generalized eigenvalue problem when the set of active constraints changes from one value of λ to the next.

3.4 Details of the Computational Algorithm

After the unconstrained problem has been solved we have an estimate of λ, say $\hat{\lambda}_o$. The algorithm to compute the constrained spline uses this value of λ as a starting point to get the estimate λ that minimizes V^c_{app}.
If $\hat{\lambda}_o = \infty$ the algorithm also requires the largest eigenvalue of the matrix $Q_2^t E^{\sigma}_{11} Q_2$ where Q_2 is obtained from the Q-R decomposition of T_1^{σ}:

$$\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} T_1^{\sigma} = \begin{pmatrix} R \\ 0 \end{pmatrix}.$$

The eigenvalue, call it ρ^*, is available from the routine that computes the unconstrained spline using Wendelberger's (1981) algorithm (see Madison Academic Computing Center, 1981).

The basic steps of the algorithm are as follows:

(1) Compute the quantities and matrices needed to solve the quadratic programming problem 3.2.

(2) Construct a regular grid of values of λ and $\hat{\lambda}_o$ in logarithmic units, in increments of 0.1 (see details at the end of the algorithm).

(3) For each value of λ do the following:

(3.1) Solve quadratic programming problem to obtain e_σ and d, using the set of active constraints at the solution for the previous value of λ as initial guess.

(3.2) Compute V_{app}^c given by (2.9).

(3.2.1) If the set of active constraints is the same as for previous value of λ go to step (3.2.3), otherwise

(3.2.2) Solve the generalized eigenvalue problem (3.6) to obtain $\rho_1,\ldots,\rho_{n+\ell-M}$

(3.2.3) Compute $V_{app}^C(\lambda)$ using (3.5) to compute the denominator.

(4) Find $\min_\lambda V_{app}^C(\lambda)$. Set $\hat{e}_\sigma = e_\sigma$ and $\hat{d} = d$ using the minimizer, $\hat{\lambda}$.

(5) Compute $[\hat{c}_\sigma{}^t : \hat{b}^t]^t = Q_2^t \hat{e}_\sigma$ to obtain the coefficients of the spline $\hat{c} = D^{-1}\hat{c}_\sigma$, $\hat{b}$ and $\hat{d}$.

In step (2) of the algorithm, the number of values of λ for which we solve the quadratic programming problem and evaluate $V^C_{app}(\lambda)$ is given as an input parameter. The user can specify the number of values to the left (n_ℓ) and to the right (n_r) of $\hat{\lambda}_o$. We recommend that n_ℓ be greater than n_r since in all our simulation studies the "minimizer" of $V^C_{app}(\lambda)$ was to the left of $\hat{\lambda}_o$.

Most of our simulations were done with $n_\ell = 15$ and $n_r = 10$. One should be careful in choosing n_ℓ and n_r because if the total number of values of λ considered is too large the computation of the spline could be very expensive. As a rule of thumb and based only on our simulation study, we would suggest considering between 15 to 20 values to the left and between 6 and 10 to the right.

The grid of values of λ is constructed as follows (in units of logarithm of λ):

If $\lambda_o < \infty$ the grid is constructed in equally spaced intervals of size 0.1, that is the grid consists of the following values:

$$\log(\hat{\lambda}_o)-0.1n_\ell,\ldots,\log(\hat{\lambda}_o)-0.1,\log(\hat{\lambda}_o),\log(\hat{\lambda}_o) + 0.1,\ldots,\log(\hat{\lambda}_o) + 0.1n_r.$$

If $\hat{\lambda}_o = \infty$ then we use the sample size n and the largest eigenvalue ρ^* from the unconstrained problem to determine an upper bound for the values of λ that will be considered. This upper bound, call it λ^* is computed as $\lambda^* = 10^3(n+k)\rho^*$ and the values of λ considered are from largest to smallest:

$$\log(\lambda^*),\ \log(\lambda^*)-2.0,\ \log(\lambda^*)-3.0,\ \log(\lambda^*)-4.0,\ (\log(\lambda^*)-4.0)-0.1,$$
$$(\log(\lambda^*)-4.0)-0.2,\ldots,(\log(\lambda^*)-4.0)-0.2(n_\ell-3).$$

In step (3.1) we use the routine QPFC to solve the quadratic programming problem.

In step (3.2.2) we use the EISPACK routines REDUC, TRED1 and TQLRAT (see Boyle, Dongarra, Garbow and Molder (1977)) to solve the generalized eigenvalue problem.

The routines DSCOMP and DSEVAL to compute and evaluate the spline are written in Ratfor (Kernighan and Plauger (1976)). The Ratfor listings are available from the first author.

All the computations are done in double precision and the routines are self-documented.

Routine DSCOMP is the routine that the user should call to solve problem (3.1).

Routine DSEVAL evaluates the spline computed by DSCOMP at a set of points in R^d. The called sequence for DSCOMP and DSEVAL as well as explanation of the variables that appear in the calling sequence are listed as comments in the source code.

As we mentioned before, the algorithm is written for the case where $L_1,\ldots,L_n$ and $N_1,\ldots,N_k$ are evaluation functionals, for example, $L_i f = f(y(i))$, $i = 1,\ldots,n$ and $N_j f = f(s(j))$, $j = 1,\ldots,k$. It is assumed that the $n + k$ points are different so that the generalized eigenvalue problem in step (3.2.2) can be solved. In the near future we plan to incorporate the handling of replicates in the algorithm. One possible strategy to handle replicates is the following: suppose that we have n_i replicates at the point y_i and denote them as $z_{i(1)},\ldots,z_{i(n)}$, then take the average

$$\bar{z}_i = \frac{\sum_{j=1}^{n} z_{i(j)}}{n_i}$$

and let $\sigma_i^2 = 1/n_i$. Then use $(\bar{z}_i, y(i))$, $i = 1,\ldots,n$ as the data with relative weights $(\sigma_1^2,\ldots,\sigma_n^2)$. The grid of points $s(1),\ldots,s(k)$ can always be chosen so that $s(j) \neq y(i) = 1,\ldots,n$, $j = 1,\ldots,k$.

In the example with real data in section 4.3 the replicates were handled using the strategy mentioned above.

4. NUMERICAL EXAMPLES: CONSTRAINED SPLINE ESTIMATES OF POSTERIOR PROBABILITIES

In this section we apply the cross-validated, constrained thin plate spline to the estimation of a bivariate posterior probability in the classification problem.

Suppose there are two populations, A_1 and A_2. If the d-dimensional random vector Y is an observation from A_1 it has the density function $f_1(y)$ and if it is from A_2 it has the density function $f_2(y)$. Let the prior probability of population A_1 be q_1. Then the posterior probability of A_1, given an observation $Y = y$, is

$$P(A_1|Y=y) = p_1(y) = q_1 f_1(y)/(q_1 f_1(y)+q_2 f_2(y)) .$$

Suppose that there is a training set of n_1 observations $Y_{11},\ldots,Y_{1n_1}$ from population A_1 and n_2 observations $Y_{21},\ldots,Y_{2n_2}$ from population A_2. We want to estimate $p(y) = p_1(y)$ given this training set.

Let $Y_1,\ldots,Y_n$, $n:=n_1 + n_2$ denote the combined sample $Y_{11},\ldots,Y_{1n_1}$; $Y_{21},\ldots,Y_{2n_2}$ from the two populations A_1 and A_2, and define the random variable

$$Z_i: \begin{matrix} 1 \text{ if } Y_i \in A_1 \\ 0 \text{ if } Y_i \in A_2 \end{matrix} .$$

Since in applications the prior probabilities q_1 and q_2 are usually unknown, we consider the estimation of

$$h = \frac{w_1 f_1}{w_1 f_1 + w_2 f_2}$$

where $w_1 = n_1/n$ and $w_2 = n_2/n$. Then if $\hat{h}$ is an estimate of h,

$$\hat{p} = \frac{(q_1/w_1)\hat{h}}{(q_1/w_1)\hat{h} + (q_2/w_2)(1-\hat{h})}$$

is an estimate of the posterior probability p.

We can think of the vector $Z = (Z_1,\ldots,Z_n)^t$ of zeros and ones as noisy observations on the values $h(y(1)),\ldots,h(y(n))$. To see this, note that, if we draw an observation Y from the density f_j with probability w_j, h = 1,2, and Z is the random variable which is 1 or 0 according as j is 1 or 2, then $E(Z|Y=y) = h(y)$.

Our estimate of h(y) is the solution h_λ to the minimization problem: find $h \in H(2,2)$ to minimize

$$\frac{1}{n} \sum_{i=1}^{n} (z_i - h(y(i))^2 + \lambda J_2(h), \tag{4.1}$$

subject to $0 \leq h(s(j)) \leq 1$, for a set $s(1),\ldots,s(k)$ to be described below, and λ is

chosen by GCV for constrained problems. Here z_i is 1 or 0 according as the observation $Y_i = y(i)$ is from population 1 or not.

In Villalobos (1983) a Monte Carlo study comparing the efficacy of this method as compared to the use of (parametric) normal theory, was carried out. When the underlying populations had f_1 and f_2 substantially different from normal the spline estimate lead to substantially improved correct classification rates, while the spline estimate is nearly as good when the two populations are normal. In this section we will only present a single example taken from that Monte Carlo study, and another example using real data. For more details and further examples, see Villalobos (1983). In the examples below we will take $q_1 = \frac{1}{2}$, $n_1 = n_2$ so that $h = p$.

We remark that iterative reweighting of the sum of squares term in (4.1) could have been implemented to take into account an estimate of the variance of z_i, but this did not lead to better estimates of p in the examples tried. Using the notation $N_2(\mu_1,\mu_2;\sigma_{11},\sigma_{22})$ for the uncorrelated bivariate normal distribution, the distributions f_1 and f_2 for the example to be presented are:

$$f_1 \sim N_2(0,0;1,1)$$

$$f_2 \sim \tfrac{1}{2} N_2(1.5,-2.5,;1,1) + \tfrac{1}{2} N_2(1.5,2.5;1,1)$$

Figure 1 gives a plot of the true posterior probability

$$p(y) \doteq p_1(y) = f_1(y)/(f_1(y) + f_2(y)).$$

Figure 2 gives a plot of a pseudo-random sample of $n_1 = 70$ observations from f_1 (crosses) and $n_2 = 70$ observations from f_2 (circles). Thus $y(1),\ldots,y(140)$ are the $n = 140$ coordinate pairs in Figure 2 and $z_1,\ldots,z_{140}$ are 1's or 0's according as the corresponding $y(i)$ came from population A_1 or A_2. These

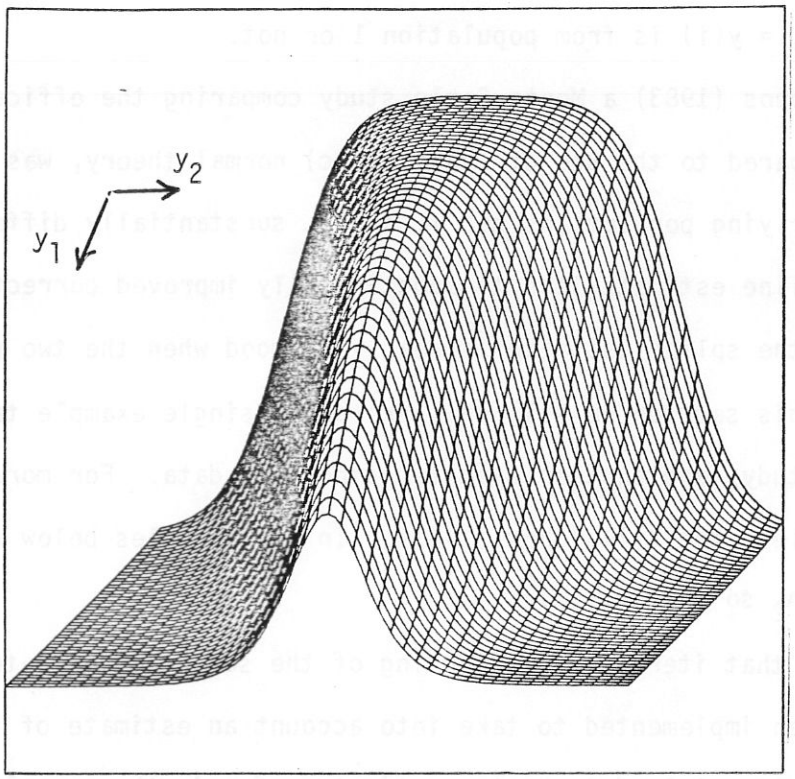

Figure 1. The true $p(y_1,y_2)$.

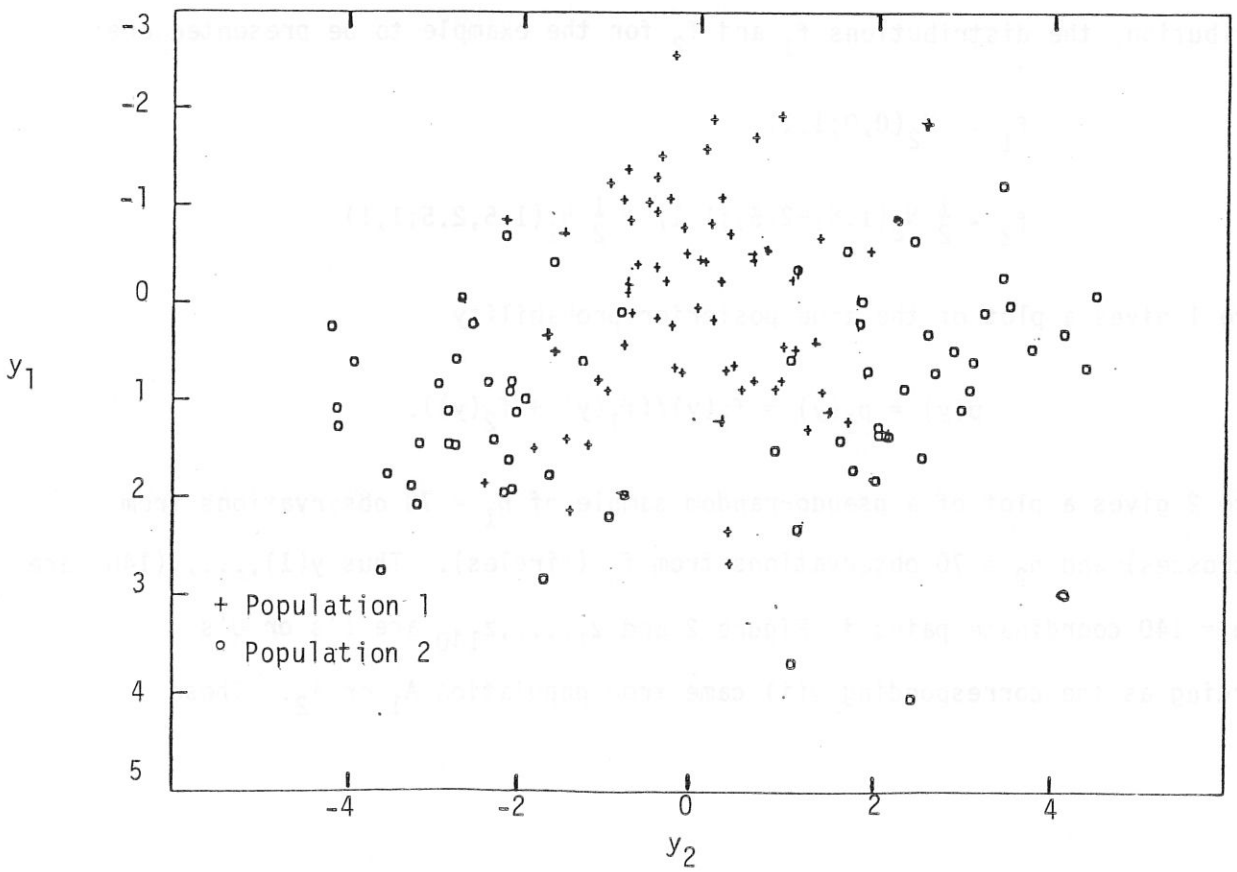

Figure 2. The pseudo data.

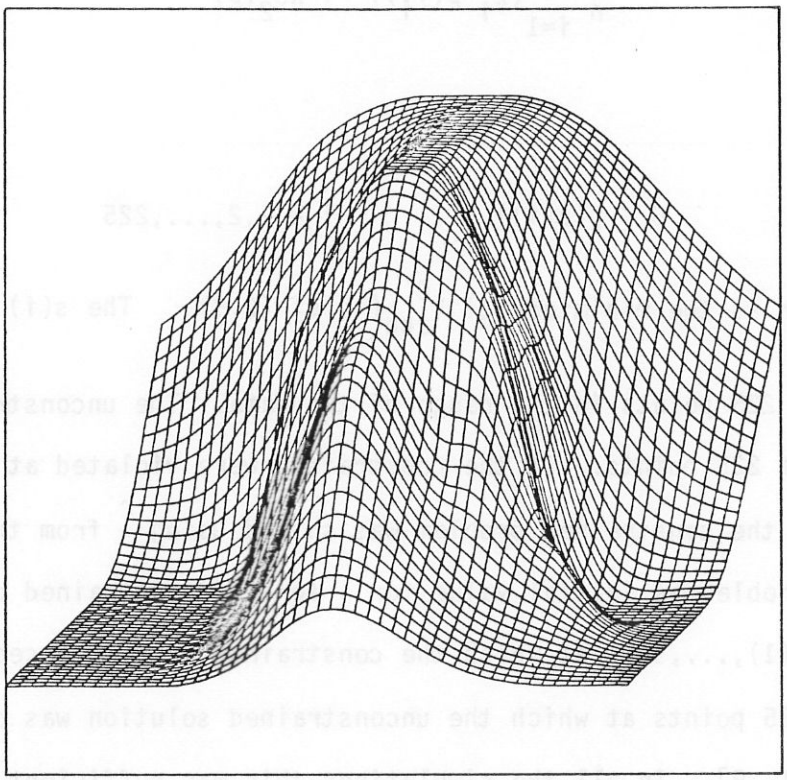

Figure 3. The constrained TP spline estimate $\hat{p}_{\lambda}(y_1,y_2)$.

observations were generated using the IMSL (1983) routines GGNSM and GGUBS. Figure 3 gives a plot of the constrained, cross-validated spline estimate $p_{\hat{\lambda}}$ of p which is the minimizer of

$$\frac{1}{n}\sum_{i=1}^{n}(z_i - p(y_i))^2 + \lambda J_2(p)$$

subject to

$$0 \leq p(s(i)) \leq 1 \qquad i = 1,2,\ldots,225$$

with $\lambda = \hat{\lambda}$ taken as the minimizer of $V_{app}^{C}(\lambda)$ of (2.9). The s(i) form a regular grid of 15×15 = 225 points in the range of the data. The unconstrained spline was evaluated at 225 points. If the constraints were violated at some of these 225 points then the constrained problem was solved using $\hat{\lambda}$ from the unconstrained problem as initial value for λ for the constrained problem. The set of points s(1),...,s(k) at which the constraints were enforced consisted of the subset of the 225 points at which the unconstrained solution was outside the interval [.1,.9]. In all the simulations this was sufficient to ensure that the resulting solution satisfied the constraints at all 225 points, and it was sufficient to restrict k to be less than 100.

Figure 4 presents a plot of the level curves of $p_{\hat{\lambda}}$ superimposed on the pseudo-data. Figure 5 presents a plot of the level curves of $p_{\hat{\lambda}}$ along with those of the true p (corresponding to Figure 1).

Figure 6 gives a plot of $V_{app}^{C}(\lambda)$. Computed values are indicated by either a "+" or a "o". These symbols are used to indicate when the set of active constraints changes. If for two consecutive values of λ the symbol changes,

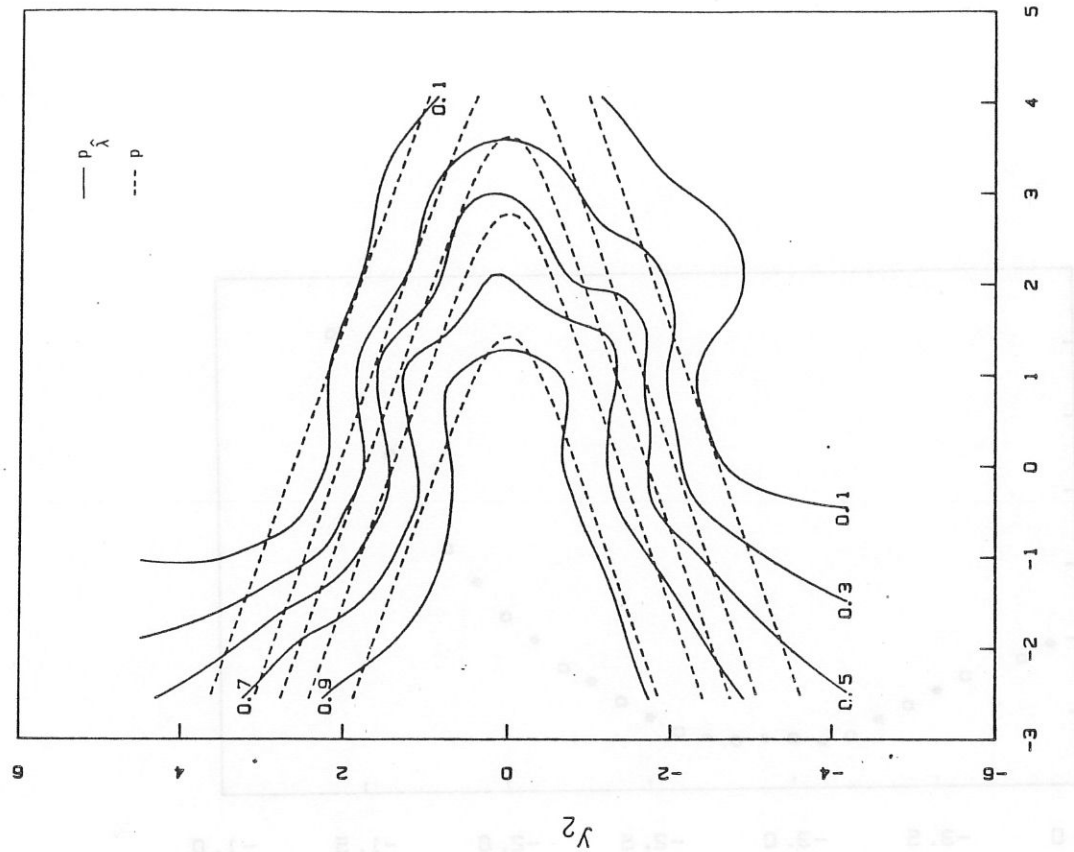

Figure 5. Level curves of $p_{\hat{\lambda}}$, and the true p.

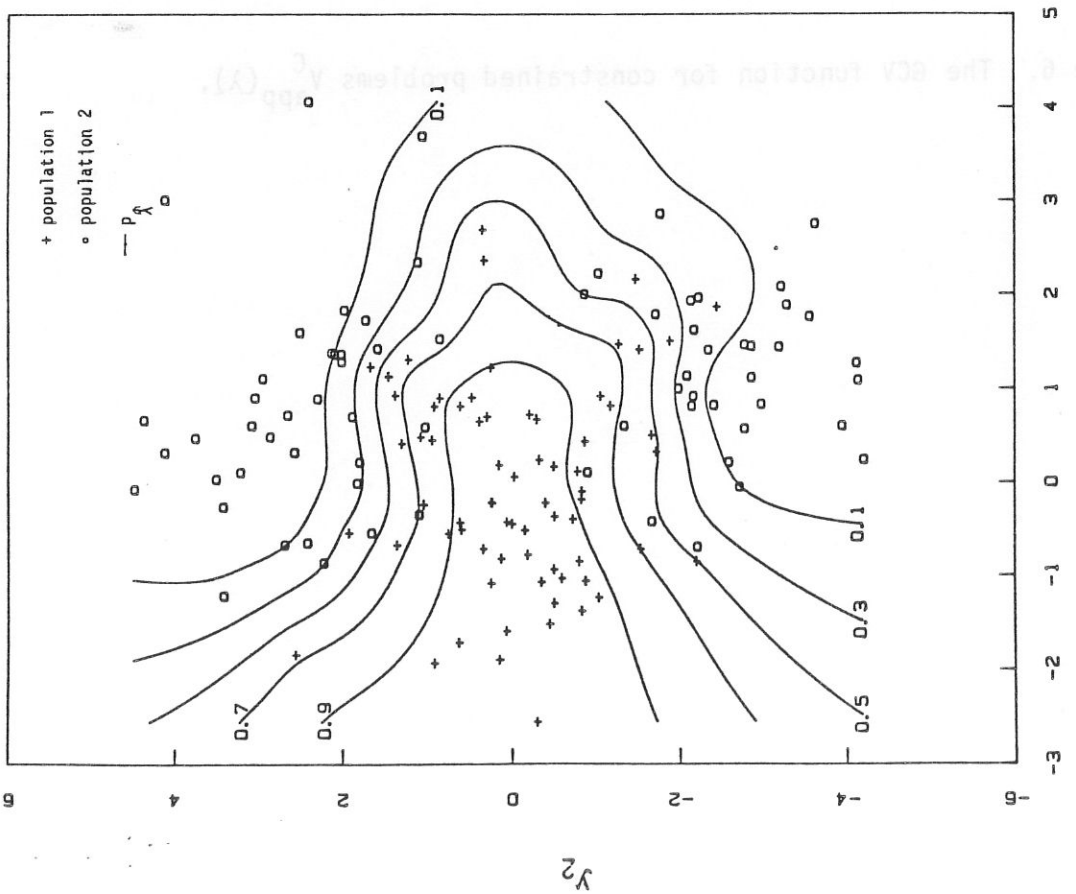

Figure 4. Level curves of $p_{\hat{\lambda}}$, and the pseudo data.

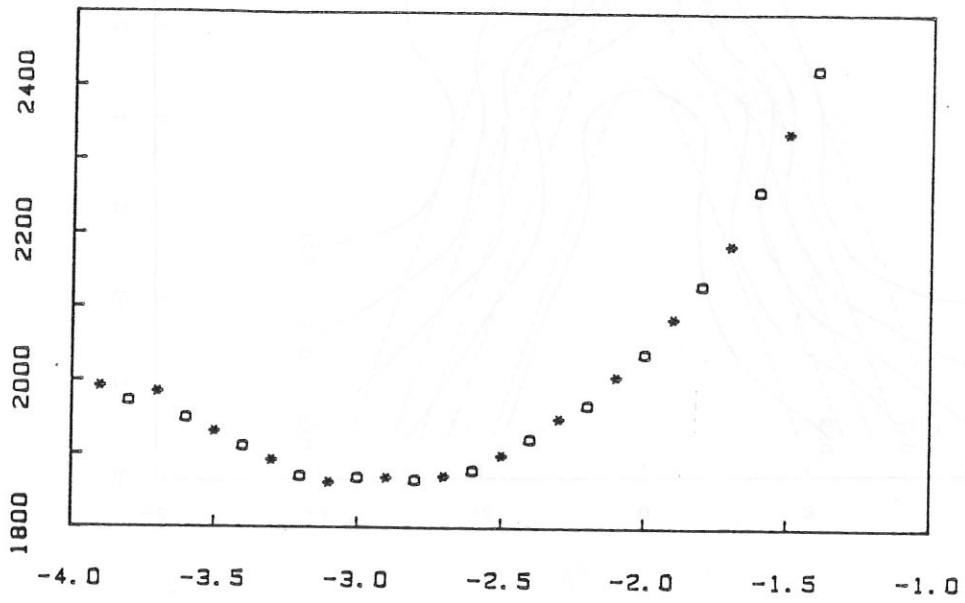

Figure 6. The GCV function for constrained problems $V^{C}_{app}(\lambda)$.

this indicates that the set of active constraints is different for those two values of λ. If values of $V^C_{app}(\lambda)$ had been computed on a finer grid, (small) jumps would have been evident when the set of active constraints changed. This did not turn out to be a serious practical problem, however.

Next we apply the method to some data from a psychological test of a group of 25 normal persons and 25 psychotics. We obtained this set of data from Smith (1947). The psychotics will be population A_1 and the normals A_2. y_1 is a score related to "size" obtained by Penrose's method (Penrose 1945) and y_2 is a score related to "shape".

Figure 7 gives a plot of the data, level curves for the spline estimate of p_λ (assuming $q_1=q_2=.5$), solid lines, and level curves for the (usual) estimate of p obtained by assuming that f_1 and f_2 are bivariate normal and estimating their parameters from the data.

Figure 8 gives $V^C_{app}(\lambda)$, with the symbols $\square$ and * having the same meaning as in Figure 6. The discontinuities are fairly apparent, but there was no practical problem in selecting the global minimum. Figure 9 shows p_λ, looking from the southeast corner of Figure 7.

We were pleasantly surprised at how good this estimate was, considering that a sum of squares of residuals is being applied to the binomial data z_i.

The penalized likelihood estimate of O'Sullivan (1983) and O'Sullivan, Yandell and Raynor (1984) is a competitor of the estimate of this chapter for the posterior probability p. These authors estimate the logit θ defined by

$$\theta = \log(p/(1-p))$$

by minimizing

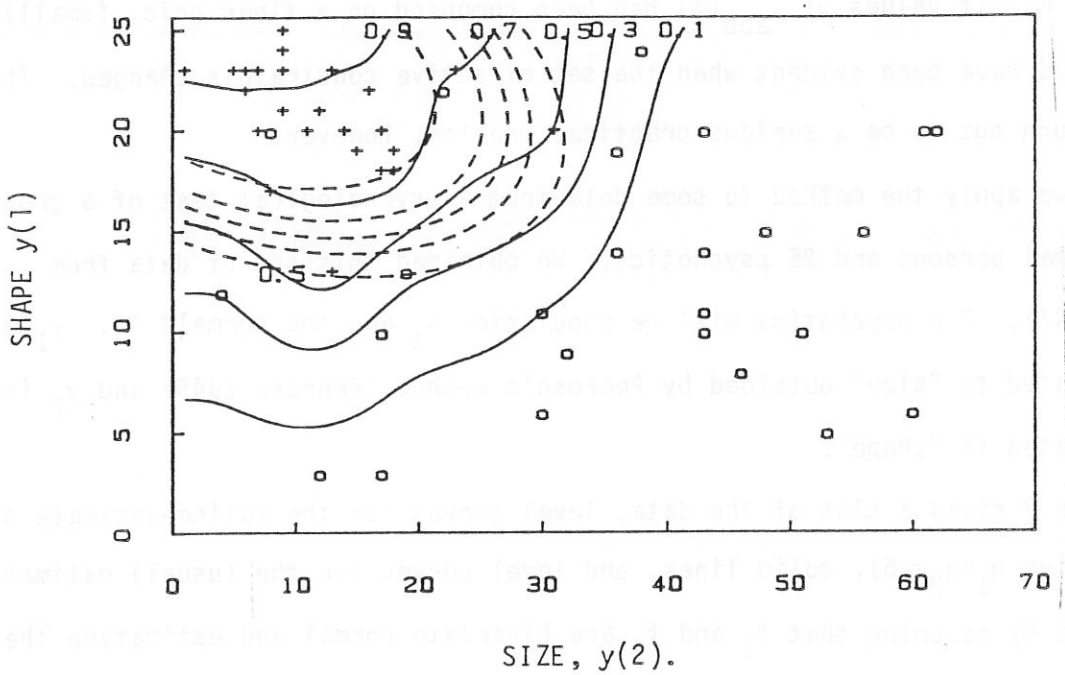

Figure 7. Constrained spline (——) and quadratic (---)
+: Psychotics; o: Normals.

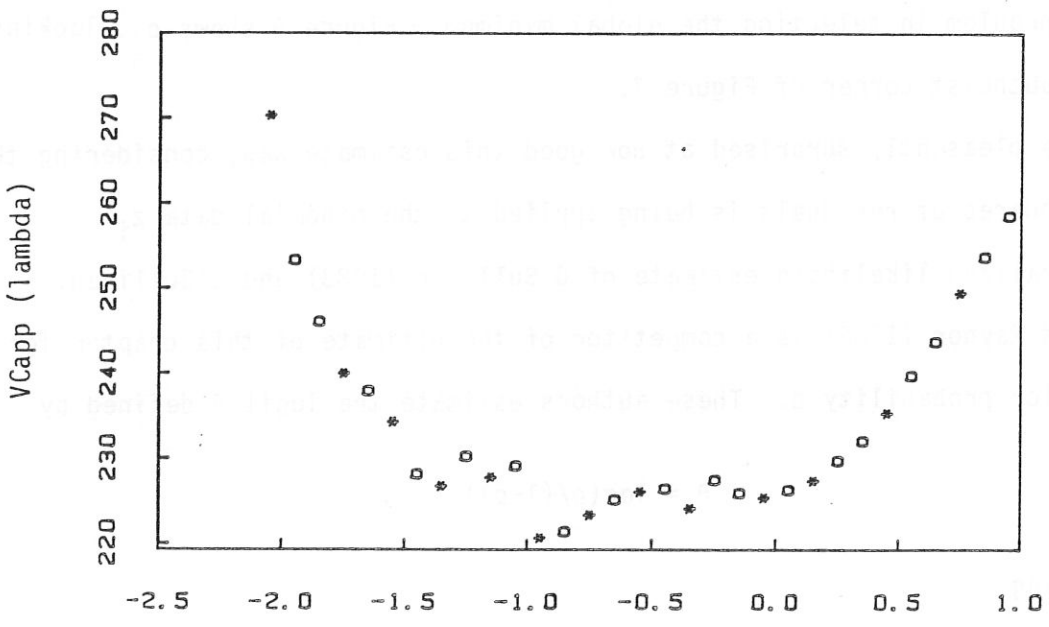

Figure 8: Approximate GCVC.

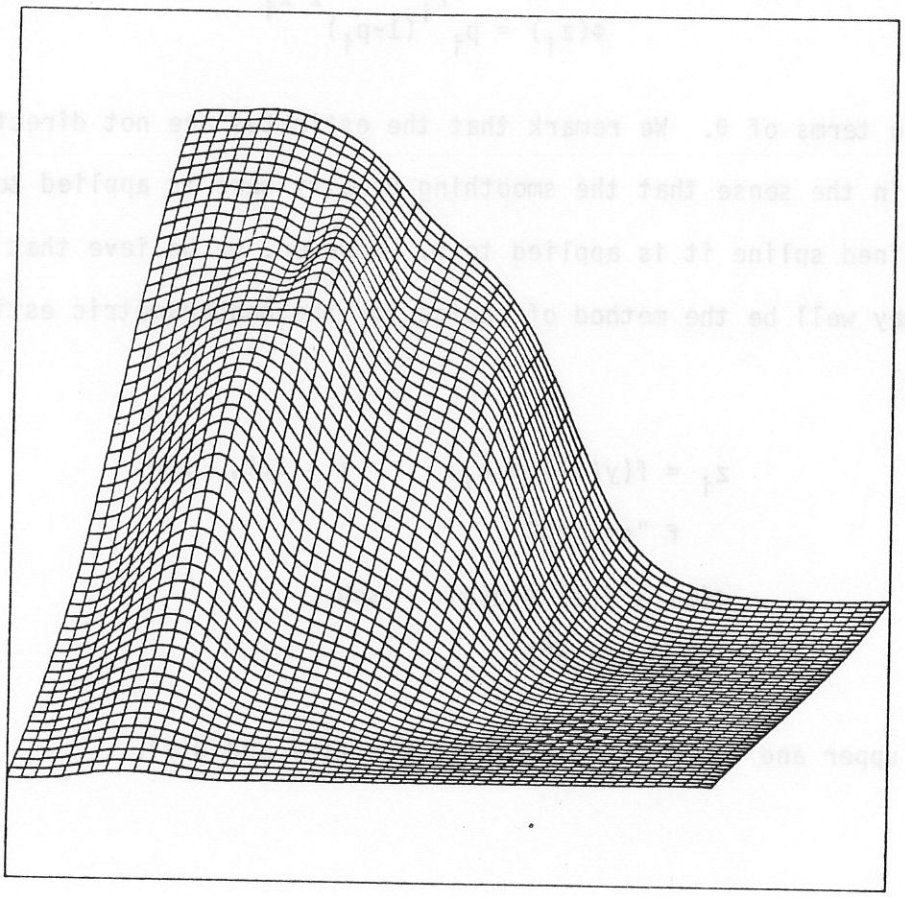

Figure 9. $p_{\hat{\lambda}}$, from the southeast corner of Figure 7.

$$Q_\lambda(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[z_i\theta(y_i - \log(1+e^{\theta(y_i)})] + \lambda J_m(\theta)$$

and using a GCV for nonlinear problems to obtain λ. The first term is just the log likelihood $\log \Phi(z_i)$ where

$$\Phi(z_i) = p_i^{z_i}(1-p_i)^{1-z_i}$$

expressed in terms of θ. We remark that the estimates are not directly comparable in the sense that the smoothing penalty here is applied to θ while in the constrained spline it is applied to p. However, we believe that the present algorithm may well be the method of choice for the nonparametric estimation of f in the model

$$z_i = f(y(i)) + \varepsilon_i, \qquad i = 1,2,\ldots,n$$

$$f \text{ "smooth"}$$

$$\ell(s) \leq f(s) \leq u(s), \quad s\varepsilon\Omega$$

$$\varepsilon \sim N(0,\sigma^2 I)$$

for smooth upper and lower bounding functions $\ell(s)$ and $u(s)$.

REFERENCES

BMDP, (1975). Biomedical Computer Programs, University of California Press, Berkeley and Los Angeles.

Bates, D. and Wahba, G. (1982). Computational methods for generalized cross-validation with large data sets. In treatment of Integral Equations by Numerical Methods, C.T.H. Baker and G.F. Miller, eds. Academic Press, London, pp. 283-296.

Bates, D. and Wahba, G. (1983). A truncated singular value decomposition and other methods for generalized cross-validation, University of Wisconsin Madison, Statistics Department Technical Report No. 715.

Boyle, J.M., Dongarra, J.J., Garbow, B.S., and Moler, C.B. (1977). Matrix eigensystem routines - EISPACK guide extension. In Lecture Notes in Computer Science, G. Boos and J. Hartmanis, eds. Springer-Verlag, New York.

Craven, P. and Wahba, G. (1979). Smoothing Noisy Data with Spline Functions, Numerische Mathematik 31, pp. 377-403.

Dongarra, J.J., Bunch, J.R., Moler, C.B., and Stewart, G.W. (1979). Linpack Users' Guide, Society for Industrial and Applied Mathematics, Philadelphia.

Duchon, J. (1976). Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces, R.A.I.R.O. Analyse Numerique 10, 12, pp. 5-12.

Dyn, N. and Wahba, G. (1982). On the estimation of functions of several variables from aggregated data, SIAM Journal of Mathematical Analysis 13, pp. 134-152.

Gill, P.E., Gould, N.I.M., Murray, W., Saunders, M.A., and Wright, M.H. (1982). Range-space methods for convex quadratic programming. T.R. SOL 82-14, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

Gill, P., Gould, N., Murray, W., Saunders, M., and Wright, M. (1984). A weighted Gram-Schmidt method for convex quadratic programming, Mathematical Programming, 30, 176-194.

Golub, G., Heath, M. and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics 31, pp. 315-324.

Hutchinson, M.F. (1984). A summary of some surface fitting and contouring programs for noisy data, Canberra, Australia: CSIRO Division of Mathematics and Statistics. (Consulting Report No. ACT 84/6).

Hutchinson, M. and Bischof, R. (1983). A new method for estimating the spatial distribution of mean seasonal and annual rainfall applied for the Hunter Valley, New South Wales, Aust. Met. Mag. 31, pp. 179-184.

IMSL, (1983). IMSL Libraries, Edition 10. (To appear.) International Mathematical and Statistical Libraries, Inc., Houston, Texas.

Kernighan, B.W. and Plauger, P.J. (1976). Software Tools, Addison-Wesley.

Kimeldorf, G. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. J. Math. Anal. Appl., 33, pp. 82-95.

Madison Academic Computing Center, (1981). Multi-dimensional spline smoothing routines, University of Wisconsin, Madison, Wisconsin.

Meinguet, J. (1979). An intrinsic approach to multivariate spline interpolation at arbitrary points. In Proceedings of the NATO Advanced Study Institute on Polynomial and Spline Approximation, B. Sahney, ed., Calgary.

O'Sullivan, F. (1983). The analysis of some penalized likelihood schemes. University of Wisconsin-Madison Statistics Department Technical Report No. 726.

O'Sullivan, F., Yandell, B., and Raynor, W. (1984). Automatic smoothing of regression functions in generalized linear models. University of Wisconsin-Madison, Statistics Department Technical Report No. 734.

Penrose, L.S. (1945). Discrimination between normal and psychotic subjects by revised examination, M. Bull. Canad. Psychol. Ass. 5, pp. 37-40.

Smith, C.A.B. (1947). Some examples of discrimination, Ann. of Eugenics 13, pp. 272-282.

Villalobos, M.A. and Wahba, G. (1983). Multivariate thin plate spline estimates for the posterior probabilities in the classification problem, Communications in Statistics, Theory and Methods, 12, No. 13, pp. 1449-1479.

Villalobos, M.A. (1983). Estimation of posterior probabilities using multivariate smoothing splines and generalized cross-validation. University of Wisconsin-Madison, Statistics Department Technical Report No. 725.

Wahba, G. (1973). On the minimization of a quadratic functional subject to a continuous family of linear inequality constraints. SIAM J. Control 11, 1, pp. 64-79.

Wahba, G. (1978). Improper priors, spline smoothing, and the problem of guarding against model errors in regression. J. Roy. Stat. Soc. B, 40, 3, 364-372.

Wahba, G. (1979). How to smooth curves and surfaces with splines and cross-validtion, Proceeding of the 24th Design of Experiments Conference, U.S. Army Research Office, Rep. 79-2, pp. 167-192.

Wahba, G. and Wendelberger, J. (1980). Some new mathematical methods for variational objective analysis using spline and cross-validation. Monthly Weather Review 108, pp. 1122-1143.

Wahba, G. (1980). Ill-posed problems: numerical and statistical methods for mildly, moderately and severely ill posed problems with noisy data. University of Wisconsin-Madison, Statistics Department Technical Report No. 595. To appear in the Proceedings of the International Conference on Ill Posed Problems, M.Z. Nashed, ed.

Wahba, G. (1982). Constrained regularization for ill-posed linear operator equations, with applications in meteorology and medicine. In Third Purdue Symposium on Statistical Decision Theory, Vol. 2, S.S. Gupta and J.O. Berger, eds., Academic Press, New York, pp. 383-418.

Wahba, G. (1983). A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. University of Wisconsin-Madison, Statistics Department Technical Report #712, submitted.

Wegman, E.J. and Wright, I.W. (1983). Splines in statistics, JASA, 78, 382, pp. 351-365.

Wendelberger, J. (1981). The computation of Laplacian smoothing splines with examples. University of Wisconsin-Madison, Statistics Department Technical Report No. 648.

Wendelberger, J. (1982). Smoothing noisy data with multi-dimensional splines and generalized cross-validation. Ph.D. Thesis, Department of Statistics, University of Wisconsin, Madison.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Technical Report No. 756
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): Inequality-Constrained Multivariate Smoothing Splines with Application to the Estimation of Posterior Probabilities

5. TYPE OF REPORT & PERIOD COVERED: Scientific Interim

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): Miguel Villalobos and Grace Wahba

8. CONTRACT OR GRANT NUMBER(s): ONR N00014-77-C-0675

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Department of Statistics, University of Wisconsin
1210 W. Dayton St.
Madison, WI 53706

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:

11. CONTROLLING OFFICE NAME AND ADDRESS:
Office of Naval Research
800 Quincy St.
Arlington, VA

12. REPORT DATE: February 1985

13. NUMBER OF PAGES: 40

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

positivity constraints;
multivariate smoothing splines

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

see attached

Inequality-Constrained Multivariate Smoothing Splines with Application to the Estimation of Posterior Probabilities

Miguel A. Villalobos
Scientific Center
IBM Mexico

Grace Wahba
Department of Statistics
University of Wisconsin

ABSTRACT

Let $z_i = f(y_1(i),y_2(i)) + \varepsilon_i$, $i = 1,2,\ldots,n$, where f is known to be a "smooth" function of (y_1,y_2) and the ε_i are independent, zero mean random variables. In addition f is known to satisfy a family of linear inequality constraints, for example $0 \leq f(y_1,y_2) \leq 1$, $(y_1,y_2)\varepsilon\Omega \subset E^2$. We propose that f be estimated as the minimizer of

$$\frac{1}{n}\sum_{i=1}^{n} w_i(z_i - f(y_1(i),y_2(i)))^2 + \lambda J_m(f)$$

subject to f satisfying the constraints. J_m is the thin plate penalty functional. A good value of λ is estimated by the method of generalized cross validation (GCV) for constrained problems. A characterization of the solution to the minimization problem with the constraints discretized is obtained from known results. We provide a numerical algorithm for computing the GCV estimate of λ and the solution to the (discretized) minimization problem. The method is applied to the estimation of posterior probabilities in the classification problem. Numerical results for both synthetic and experimental data are given.

Key Words: thin plate splines; inequality constraints; cross validation; constrained surface estimation

This research supported by the Office of Naval Research under Grant No. N00014-77-C-0675 and by NASA under Grant No. NAG5-316.