DEPARTMENT OF STATISTICS
University of Wisconsin
1210 West Dayton St.
Madison, WI 53706

TECHNICAL REPORT NO. 899

January 20, 1993

Soft Classification, a.k.a. Risk Estimation, via Penalized Log Likelihood and Smoothing Spline Analysis of Variance

by

Grace Wahba, Chong Gu, Yuedong Wang and Rick Chappell

Soft Classification, a. k. a. Risk Estimation, via Penalized Log Likelihood and Smoothing Spline Analysis of Variance

Grace Wahba, Chong Gu, Yuedong Wang and Richard Chappell*

January 20, 1993

Abstract

We discuss a class of methods for the problem of 'soft' classification in supervised learning. In 'hard' classification, it is assumed that any two examples with the same attribute vector will always be in the same class, (or have the same outcome), whereas in 'soft' classification, two examples with the same attribute vector do not necessarily have the same outcome, but the *probability* of a particular outcome does depend on the attribute vector. In this paper we will describe a family of methods which are well suited for the estimation of this *probability*. The method we describe will produce, for any value in a (reasonable) region of the attribute space, an estimate of the *probability* that the next example will be in class 1. Underlying these methods is an assumption that this probability varies in a smooth way (to be defined) as the predictor variables vary. The method combines results from Penalized log likelihood estimation, Smoothing splines, and Analysis of variance to get the PSA class of methods. In the process of describing PSA we discuss some issues concerning the computation of degrees of freedom for signal, which has wider ramifications for the minimization of generalization error in machine learning. As an illustration we apply the method to the Pima-Indian Diabetes data set in the UCI Repository, and compare the results to Smith *et al*(1988) who used the ADAP learning

*This research was supported by NSF Grants DMS9121003 and DMS-9101730, NEI Grant R01 EY09946-01, and the Wisconsin Alumni Research Foundation. Parts of this paper were presented by the first author in invited talks at the Santa Fe Institute Workshop on Supervised Machine Learning, Santa Fe, NM, August 6-7, 1992, and at the Third Annual Workshop on Computational Learning Theory and 'Natural' Learning Systems, Madison WI, August 27-29, 1992. ©Grace Wahba 1993. e-mail wahba@stat.wisc.edu, chong@pop.stat.purdue.edu, wang@stat.wisc.edu and chappell@stat.wisc.edu. To appear in the proceedings of the Santa Fe Workshop, D. Wolpert and A. Lapedes, eds., Addison-Wesley

algorithm on this same data set to forecast the onset of diabetes mellitus. If the probabilities we obtain are thresholded to make a hard classification to compare with the hard classification of Smith *et al*(1988), the results are very similar, however, the intermediate probabilities that we obtain provide useful and interpretable information on how the risk of diabetes varies with some of the risk factors.

KEY WORDS: soft classification, hard classification, penalized log likelihood estimation, risk factor estimation, RKPACK, smoothing splines, analysis of variance, degrees of freedom for signal, cross validation, unbiased risk estimate.

1 Introduction to 'soft' classification and the bias-variance trade-off

A typical problem in medical data analysis is the following: Records of attribute vectors as well as records of the outcome for each example (patient) for n examples are available as training data. Based on the training data, it is desired to predict the outcomes, for any new examples that may be presented in the future with only their attribute vectors. In this paper we will consider only two outcomes (1 and 0), where 1 indicates that a particular medical condition of interest was (later) found to be true, and 0 indicates that it was found not to be true. As a concrete example, O'Sullivan, Yandell and Raynor(1986)(OYR) consider records from the Western Electric Health study which gave patient blood pressure and cholesterol level at the start of the study, and an indicator 1 or 0 indicating that the patient did or did not have a heart attack in the 19 year follow up period. Assuming that the attribute pair of blood pressure and cholesterol level has been suitably scaled to a rectangle $\mathcal{T}$, the 'hard' classification problem would be to partition $\mathcal{T}$, or, more precisely, some subregion of interest of it, into two non-overlapping regions, one labeled '1', and the other labeled '0'. If a neural network (NN) is used for this task, the partition is generally not made explicit, however, if a new example is presented to the (trained) NN, the NN will produce a 1 or a 0 according to which region the attribute vector for the new example lies. In 'soft' classification, the desired (trained) algorithm will produce not a 1 or 0 but a value p (usually strictly) between 1 and 0, which is an estimate of the probability that the new example is, or will be a '1'. OYR is a prototype for the 'soft' classification method that we will be describing. In order to do 'soft' classification by the methods we will be discussing, we will be assuming that the desired probability varies 'smoothly' with any continuous attribute (predictor variable). Categorical predictor variates will (later) be allowed, and if there are more than a few categories, some 'smoothness' penalties on the categorical values will be required. We remark that to talk about probabilities we should carefully construct a 'worldview' in which such probabilities make unambiguous sense, and we shall do that later.

It is well known that smoothness penalties and Bayes estimates are intimately related (see, for example Kimeldorf and Wahba(1970, 1971), Wahba(1978), Wahba(1990), Buntine and Weigend(1991)). We will not discuss this further in the present paper except to note that our philosophy with regard to the use of priors in Bayes estimates is to use them to generate reasonable penalty functionals with appropriate free parameters to generate a structure on which estimates can be 'hung', (see Wahba(1990) Chapter 3, also Wahba(1992)) and then use cross-validation (CV), generalized cross validation (GCV), unbiased risk estimation (UBR) or some other *performance oriented* method to choose the free (regularization) parameter(s) in the penalty functional to minimize some computable proxy for the generalization error (a.k.a. the bias-variance tradeoff, see Geman, Bienenstock and Doursat(1992).) A person who completely believed the associated prior might use maximum likelihood to choose the free parameters, but maximum likelihood may not be robust against an unrealistic prior (that is, it may not do very well from the generalization point of view if the prior is not completely up to snuff), see Wahba(1985). Another proposal frequently put forward is to assign a hyperprior to the free parameters. However, except in particular cases where much is known *a priori*, there is no reason to believe that the use of hyperpriors will beat out a performance oriented criterion which is a good proxy for the generalization error, assuming, of course, that low generalization error is the true goal.

The 'soft' classification structure that we will describe in this paper historically begins with the penalized log likelihood risk estimate of O'Sullivan (1983) and OYR, which was extended by Gu(1990) in such a way that penalized log likelihood risk estimation could be combined with smoothing spline analysis of variance (SS-ANOVA) as described by Wahba(1986, 1990), Gu(1989), Gu, Bates, Chen and Wahba(1989), Chen, Gu and Wahba(1989), Gu and Wahba(1991a,b, 1993a,b). The SS-ANOVA allows a variety of interpretable structures for the possible relationships between the predictor variables and the outcome. Other recent work with the same goal but with technically different approaches than that described here include Gray(1992) and Tibshirani and

LeBlanc(1992). For an informative overview of this area from a statistician's point of view, see Ripley(1992).

Gu(1992b) has brought to the fore some rather subtle issues concerning the implementation of GCV and UBR in choosing (possibly multiple) smoothing parameters in the context of non-Gaussian (that is, non quadratic) log likelihoods. Both of these estimates require the calculation of the *degrees of freedom for signal (df-signal).* We claim that df-signal is going to be a key quantity in any method which does not seek to fit the data exactly, since it is intimately related to how close the model reproduces the training data. In this paper we will review and discuss some of the results in Gu(1992b) and apply them to a Penalized log likelihood Smoothing spline Analysis of variance (PSA) model for 'soft' classification. Our discussion concerning df-signal below is related to an intriguing proposal of Moody (1991), and has possible ramifications in other structures for machine learning, with respect to the calculation of df-signal in the general case where there is a non-quadratic optimization problem to be solved numerically. We will in our example use a UBR method (see Craven and Wahba(1979) and references cited there) modified for the binomial case and implemented by the self-voting algorithm in Gu(1992b), where these subtle issues are discussed in some detail.

In Section 2 we review the penalized log likelihood estimate described in OYR and use that as a simple vehicle to describe the bias-variance tradeoff (a. k. a. generalization error). In Section 3 we discuss the above mentioned subtle issues in choosing the smoothing parameters and the key quantity df-signal. In Section 4 we describe the general PSA model, and discuss how to compute it. In Section 5 we apply several PSA models to the estimation of the risk of diabetes mellitus, from the Pima-Indian data set in the UCI Repository of Machine Learning Databases. We compare the best of the PSA models to the use of the ADAP NN classification algorithm as applied by Smith*et al*(1988) to the same Pima-Indian data set.

2 Soft classification and penalized log likelihood risk factor estimation

First, to describe the 'worldview' adopted in this paper, let t be a vector of attributes, $t \in \Omega \in \mathcal{T}$, where Ω is some region of interest in attribute space $\mathcal{T}$. We imagine that the 'world' consists of an arbitrarily large population of potential examples, whose attribute vectors are distributed in some way over Ω and furthermore, considering all members of this 'world' with attribute vector in a small neighborhood about t, the fraction of them that are 1's is $p(t)$. We are implicitly assuming that small neighborhoods about t can be defined.

Our training set is assumed to be a random sample of n examples from this population, whose classification is known, and our goal is to be able to estimate $p(t)$ for any $t \in \Omega$. In 'soft' classification, we do not expect classification to be a 'sure thing', that is we do not expect $p(t)$ to be 0 or 1 for large portions of Ω. Here is how we would use our estimate $\hat{p}(t)$ of $p(t)$ - if our classification is the presence or absence of some medical condition, then, when a new patient (example) appears, with attribute vector t, we announce that their risk of getting the medical condition is approximately $\hat{p}(t)$, furthermore (in some situations) we can also announce that if they change their attribute vector to t', then they can, approximately, change their risk to $\hat{p}(t')$. A 'hard' classification can be done if desired, by thresholding at a fixed value of $\hat{p}$, however in medical applications in particular, the 'patient' would probably prefer knowing $\hat{p}$, rather than just which side of the threshold that they fell in. For example, the probability $\hat{p}$ may be useful in suggesting the urgency of treatment. The concept of a probability is also useful in large demographic studies, where one may wish to estimate what proportion of a population will later develop some medical condition.

We now review the penalized log likelihood risk estimate in O'Sullivan(1983) and OYR. First, define the logit $f(t)$ by $f(t) = \log[p(t)/(1-p(t))]$, the logit provides a convenient and commonly used means of transforming the unit interval into the real line, see for example McCullagh and Nelder(1989). In OYR, t is a vector containing two continuous variates, $t = (t_1, t_2)$, and the logit f is assumed to be 'smooth' as a function of these continuous variates, in the sense of possessing square integrable second derivatives. More precisely OYR assume that f is in an appropriately defined collection of functions for which the thin plate spline 'smoothness penalty' $J(f)$ defined by

$$J(f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\frac{\partial^2 f}{\partial t_1^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial t_1 \partial t_2}\right)^2 + \left(\frac{\partial^2 f}{\partial t_2^2}\right)^2 dt_1 dt_2 \tag{2.1}$$

is well defined and finite. (See Wahba and Wendelberger(1980), or Wahba(1990) and references cited there for further information concerning this penalty functional.) The OYR estimate $p_\lambda(t)$ of $p(t)$ is then obtained, for any fixed non-negative λ, as the minimizer, in the above mentioned collection of functions, of

$$- \, log \;\; likelihood\{data, f\} + \frac{n}{2}\lambda J(f). \tag{2.2}$$

The log likelihood is defined as follows: Let the training data be $\{y_i, t(i), i = 1, ...n\}$ where y_i has the value 1 or 0 according to the classification of example i, and $t(i) = (t_1(i), t_2(i))$ be the attribute vector for example i. If the n examples are a random sample from our 'world', then the likelihood of this data, given $p(\cdot)$, is (with some abuse of notation)

$$likelihood\{y, p\} = \Pi_{i=1}^n p(t(i))^{y_i}(1 - p(t(i)))^{1-y_i}, \tag{2.3}$$

which is the product of n binomial likelihoods. By substituting in f and taking logs, we have

$$- \, log \;\; likelihood\{y, f\} \equiv \mathcal{L}(y, f) = \sum_{i=1}^{n} \log(1 + e^{f(t(i))}) - y_i f(t(i)). \tag{2.4}$$

The minimizer, f_λ, of

$$\mathcal{L}(y, f) + \frac{n}{2}\lambda J(f) \tag{2.5}$$

will be taken as the estimate of $f(t)$. Assuming that the $t(i) = (t_1(i), t_2(i))$ do not fall on a straight line, f_λ is known to be in the n dimensional space of functions with a representation

$$f_\lambda(t) = d_0 + d_1 t_1 + d_2 t_2 + \sum_{i=1}^{n} c_i |t - t(i)|^2 log|t - t(i)|, \tag{2.6}$$

where $|t - t(i)|$ is the Euclidean distance between t and $t(i)$ and the $\{c_i\}$ satisfy the three conditions $0 = \sum_{i=1}^n c_i = \sum_{i=1}^n c_i t_1(i) = \sum_{i=1}^n c_i t_2(i)$, see Wahba(1990), Wahba and Wendelberger(1980) and references cited there. A Newton-Raphson iteration can be used to compute the coefficients. The likelihood function $\mathcal{L}$ will be maximized if $p(t(i))$ is 1 or 0 according as y_i is 1 or 0, hence, the reader may convince theirself that as $\lambda \to 0$, f_λ must tend to $+\infty$ or $-\infty$ at the data points. Thus, by letting λ be small, we can come close to fitting the data points, but it is fairly clear that unless the 1's and 0's are well segregated in attribute space, f_λ will be a very 'wiggly' function and the generalization error (which we have not exactly defined yet) is likely to be large. For the moment, think of the generalization error as a failure of $\hat{p}(t) \equiv p_\lambda(t)$ to adequately approximate $p(t)$ according to some meaningful criterion. If λ is very large, it can be shown that f_λ will tend to a linear function of the components of t. Then, unless the true logit function is of this form, the generalization error can be expected to be large. We note that it is common in medical data analysis to fit a parametric model to the data in which the logit is linear in the attribute vector components, (see, for example McCullagh and Nelder(1989)). The choice of λ here represents a tradeoff between overfitting and underfitting the data, and this is the soft classification version of the bias-variance tradeoff discussed in Geman *et al*(1992). In practice, it will generally be very important to obtain a good value of λ. Before proceeding, then, we should decide what we mean by a good value of λ. Given the family $p_\lambda, \lambda \geq 0$, we want to choose λ so that p_λ is close to the 'true' but unknown p in some sense. Then, if a large number of new examples arrive with attribute vector in a neighborhood of t, $p_\lambda(t)$ will be a good estimate of the fraction of them that are 1's. 'Closeness' can be defined in a number of reasonable ways, for example as the norm of the difference between p_λ and p in some function space. In this paper we will primarily use the Kullbach-Leibler distance $KL_\nu(p_\lambda, p)$, a commonly used criterion sometimes appearing in the NN literature under other names. (Note that it is not a real distance.) We will mention other criteria later. If $\nu(t)$ is some probability measure on $\mathcal{T}$, define $KL_\nu(p_\lambda, p)$ with respect to ν as

$$KL_\nu(p_\lambda, p) = \int \left[p(t) log\left(\frac{p_\lambda(t)}{p(t)}\right) + (1 - p(t)) \log\left(\frac{1 - p_\lambda(t)}{1 - p(t)}\right)\right] d\nu(t). \tag{2.7}$$

$KL_\nu(p_\lambda, p)$ as a measure of closeness of p_λ to p reflects the following 'game': Nature chooses a new example with attribute t according to the probability distribution $\nu(t)$. Then the computer scientist-statistical data analyst (cs-sda) announces that the probability of a 1 for this example is $p_\lambda(t)$. Nature now chooses the outcome for this example as a 1 with probability $p(t)$ and 0 with probability $1 - p(t)$. If the example turns out to be a 1 the 'loss' to the cs-sda is $\log\left(\frac{p_\lambda(t)}{p(t)}\right)$ and if the example is a 0, then the 'loss' is $\log\left(\frac{1-p_\lambda(t)}{1-p(t)}\right)$. Thus the expected loss is

$$p(t) log\left(\frac{p_\lambda(t)}{p(t)}\right) + (1 - p(t)) \log\left(\frac{1 - p_\lambda(t)}{1 - p(t)}\right) \tag{2.8}$$

and averaging over the distribution ν of the t's gives (2.7). Note that the expected loss is minimized if $p_\lambda = p$. Since KL_ν is not computable from the data, it is necessary to develop a computable proxy for it. By a computable proxy is meant a function of λ that can be calculated from the training set which has the property that its minimizer is a good estimate of the minimizer of KL_ν. Note that to minimize KL_ν, it is only necessary to minimize

$$\int \left[p(t) \log(p_\lambda(t)) + (1 - p(t)) \log(1 - p_\lambda(t))\right] d\nu(t) \tag{2.9}$$

over λ since (2.7) and (2.9) differ by something that does not depend on λ. Leaving-out-half cross validation $1/2CV$ is one conceptually simple and generally defensible (albeit possibly wasteful) way of choosing λ to minimize a proxy for $KL_\nu(p_\lambda, p)$. The n examples are randomly (important!) divided in half and the first $n/2$ examples are used to compute p_λ for a series of trial values of λ. Recall that, since p_λ has a representation in terms of a set of basis functions, once the coefficients have been computed $p_\lambda(t)$ can be evaluated relatively cheaply for any attribute vector t in Ω. Then, the remaining $n/2$ examples are used to compute

$$\begin{aligned}\widehat{KL}_{1/2CV}(\lambda) &= \frac{2}{n}\sum_{i=n/2+1}^{n}[y_i \log p_\lambda(t(i)) + (1-y_i)log(1-p_\lambda(t(i)))] \\ &= \frac{2}{n}\sum_{i=n/2+1}^{n}[y_i f_\lambda(t(i)) - \log(1+e^{f_\lambda(t(i))})] \qquad (2.10)\end{aligned}$$

for the trial values of λ. Since the expected value of y_i is $p(t(i))$, (2.10) is, for each λ an unbiased estimate of (2.9) with $d\nu$ the sampling distribution of $t(\frac{n}{2}+1), ..., t(n)$. The parameter λ would then be chosen by minimizing (2.10) over the trial values. Note that it is inappropriate to just evaluate (2.10) using the same data that was used to obtain f_λ, as that would lead to overfitting the data. Variations on (2.10) are obtained by successively leaving out groups of data. A repeated leaving-out-one (or ordinary cross validation(OCV)) proxy for $\widehat{KL}_\nu(p_\lambda, p)$ would go as follows: Let $f_\lambda^{[k]}$ be the estimate of f_λ (i. e. the minimizer of (2.5)) with the kth data point left out. Then the OCV proxy for KL_ν is

$$\widehat{KL}_{OCV}(\lambda) = \frac{1}{n}\sum_{k=1}^{n}[y_k f_\lambda^{[k]}(t(k)) - \log(1+e^{f_\lambda^{[k]}(t(i))})] \qquad (2.11)$$

and λ is chosen to minimize (2.11). Essentially this estimate was suggested by Cox and Chang(1990) in the case of a single predictor variable with $J(f) = \int(f''(t))^2dt$. In this case f_λ is a cubic spline, and Cox and Chang proposed a computational algorithm which used special computationally efficient methods available for polynomial splines. While OCV represents a relatively efficient use of the data, the computation required is likely to be expensive in general. In the next Section we will describe approximate GCV and UBR proxies for (2.9), which we have been able to compute in more complicated situations.

3 Df-signal and the GCV and UBR estimates for λ

In order to understand the GCV and UBR estimates we will describe in the 'soft' classification context, we will first describe their role in a simpler setup, namely when $\mathcal{L}$ is quadratic in the data and the unknown. This is the situation where we have n examples $\{y_i, t(i), i = 1, 2, ..., n\}$, where our 'world view' says that nature chooses t from the distribution ν, and then y_i is related to f by

$$y_i = f(t(i)) + \epsilon_i, \quad i = 1, ..., n, \tag{3.1}$$

where f is assumed to be smooth as before, and the ϵ_i are assumed to be independent, zero mean Gaussian random variables with mean 0 and common, possibly unknown variance σ^2. It is desired to estimate f from this data. The estimate f_λ is the minimizer of

$$\sum_{i=1}^{n}(y_i - f(t(i)))^2 + n\lambda J(f). \tag{3.2}$$

These estimates are discussed in Wahba(1990) and in many of the 13 pages of references cited there, see also Wahba (1992). There is a so called smoother matrix $A(\lambda)$ defined by the property

$$\begin{pmatrix} f_\lambda(t(i)) \\ \vdots \\ f_\lambda(t(n)) \end{pmatrix} = A(\lambda)y. \tag{3.3}$$

Smoother matrices are symmetric nonnegative definite and have all their eigenvalues in the interval $[0, 1]$. By analogy with ordinary regression, the trace of $A(\lambda)$ is known as the degrees of freedom for signal, (df-signal) see Wahba(1983), Buja, Hastie and Tibshirani(1989).

The OCV estimate of λ in this context was suggested by Wahba and Wold(1975), we review it here to show its relationship to GCV (generalized cross validation). Letting $f_\lambda^{[k]}$ be the minimizer of (3.2) with the kth data point left out, the OCV estimate of λ is the minimizer of $V_O(\lambda)$ defined by

$$V_O(\lambda) = \frac{1}{n}\sum_{k=1}^{n}(y_k - f_\lambda^{[k]}(t(k)))^2 \tag{3.4}$$

and the celebrated 'leaving-out-one' lemma (a proof is in Wahba(1990)) gives the identity

$$V_O(\lambda) \equiv \frac{1}{n}\sum_{k=1}^{n}(y_k - f_\lambda(t(k)))^2/(1 - a_{kk}(\lambda))^2. \tag{3.5}$$

where a_{kk} is the kkth entry of $A(\lambda)$. The GCV estimate of λ is the minimizer of $V(\lambda)$ defined by

$$V(\lambda) \;=\; \frac{1}{n}\sum_{k=1}^{n}(y_k - f_\lambda(t(k)))^2/(1 - \frac{1}{n}\sum_{\ell=1}^{n}a_{\ell\ell}(\lambda))^2 \tag{3.6}$$

$$\equiv \frac{\frac{1}{n}||(I - A(\lambda))y||^2}{\left(\frac{1}{n}tr(I - A(\lambda))\right)^2}. \tag{3.7}$$

Both $V_0(\lambda)$ and $V(\lambda)$ are proxies for the criterion $R(\lambda)$ defined by

$$R(\lambda) = \int (f_\lambda(t) - f(t))^2 d\nu(t) \tag{3.8}$$

in the sense that the minimizers of $V(\lambda)$ and $V_0(\lambda)$ are good estimates of the minimizer of the expected value of $R(\lambda)$, with $V(\lambda)$ having superior theoretical properties under certain circumstances, and much superior computational properties. See Craven and Wahba(1979), Li(1985, 1986). The expected value is here taken over the random variables ϵ_i. The UBR estimate in this context was proposed by Craven and Wahba (1979) based on Mallows celebrated C_p, Mallows(1973). This estimate requires the knowledge of, or a good estimate of σ^2, and is the minimizer of

$$U(\lambda) = \frac{1}{n}||(I - A(\lambda))y||^2 + 2\frac{\sigma^2}{n}trA(\lambda). \tag{3.9}$$

In what follows we will take the sampling distribution for the $\{t(i)\}$ as a proxy for $d\nu$ of (3.8). Then $U(\lambda)$ is a proxy for $R(\lambda)$ in that

$$EU(\lambda) = ER(\lambda) + \sigma^2. \tag{3.10}$$

We include a short proof since it is so simple: Let $f = (f(t(1)), ..., f(t(n)))', \epsilon = (\epsilon_1, ..., \epsilon_n)'$, then

$$\begin{aligned} EU(\lambda) &= E\frac{1}{n}||(I - A(\lambda))(f + \epsilon)||^2 + 2\frac{\sigma^2}{n}trA(\lambda) \\ &= \frac{1}{n}||(I - A(\lambda))f||^2 + \frac{\sigma^2}{n}tr(I - A(\lambda))^2 + 2\frac{\sigma^2}{n}trA(\lambda) \\ &= \frac{1}{n}||(I - A(\lambda))f|| + \frac{\sigma^2}{n}trA^2(\lambda) + \sigma^2 \\ &= ER(\lambda) + \sigma^2. \end{aligned}$$

We note that a crude version of the argument supporting the properties of $V(\lambda)$ as a proxy for $R(\lambda)$ in the case that σ^2 is not known goes as follows (see Craven and Wahba(1979) for more details):

$$EV(\lambda) = [\frac{1}{n}||(I - A(\lambda))f||^2 + \sigma^2 tr(I - A(\lambda))]/(1 - \frac{1}{n}trA(\lambda))^2. \tag{3.11}$$

Assuming that $\frac{1}{n}trA(\lambda)$ is small compared to 1 in the neighborhood of the optimal λ (a condition insuring that this is true is necessary for GCV to work well) then

$$\begin{aligned} EV(\lambda) &\sim [\frac{1}{n}||(I - A(\lambda))f||^2 + \frac{1}{n}\sigma^2 tr(I - A(\lambda))^2][1 + \frac{2}{n}trA(\lambda) + ...] \\ &\sim [\frac{1}{n}||(I - A(\lambda)f||^2 + \frac{\sigma^2}{n}trA^2(\lambda) + \sigma^2][1 + o(1)] \\ &\sim [ER(\lambda) + \sigma^2][(1 + o(1)]. \end{aligned}$$

Now, let us return to the case of soft classification, where $\frac{1}{2}\sum_{i=1}^{n}(y_i - f(t(i)))^2$ which is a multiple of the Gaussian log likelihood, is replaced by a log likelihood of the general form

$$\mathcal{L}(y,f) = \sum_{i=1}^{n}[b(f(t(i)) - y_i f(t(i)))], \tag{3.12}$$

where $b(\cdot)$ is given. In the binomial case that we are discussing here $b(f) = log(1+e^f)$, but many likelihood functions can be represented this way. See McCullagh and Nelder(1989). For future use we note that in the binomial case it is easy to verify that

$$Ey_i \;=\; p(t(i)) = \frac{e^{f(t(i))}}{1+e^{f(t(i))}} = b'(f(t(i))) \tag{3.13}$$

$$vary_i \;=\; p(t(i))(1-p(t(i))) = \frac{e^{f(t(i))}}{(1+e^{f(t(i))})^2} = b''(f(t(i))), \tag{3.14}$$

however, these relations between the mean and variance of y_i and the first and second derivatives of b hold for any log likelihood of the form (3.12). Representing f either exactly by using a basis for the space of functions in (2.6), or approximately by suitable basis functions write

$$f \simeq \sum_{k=1}^{N} c_k B_k \tag{3.15}$$

then we need to find $c = (c_1, \ldots, c_N)'$ to minimize

$$I_\lambda(c) = \sum_{i=1}^{n} b(\sum_{k=1}^{N} c_k B_k(t(i))) - y_i(\sum_{k=1}^{N} c_k B_k(t(i))) + \frac{n}{2}\lambda c'\Sigma c, \tag{3.16}$$

where Σ is the necessarily non-negative definite matrix determined by $J(\sum_k c_k B_k) = c'\Sigma c$. Straightforward calculations show that the gradient $\bigtriangledown I_\lambda$ and the Hessian $\bigtriangledown^2 I_\lambda$ of $I_\lambda$are given by

$$\bigtriangledown I_\lambda = \begin{pmatrix} \frac{\partial I_\lambda}{\partial c_1} \\ \vdots \\ \frac{\partial I_\lambda}{\partial c_N} \end{pmatrix} \;=\; X'(p_c - y) + n\lambda\Sigma c, \tag{3.17}$$

$$\left\{\bigtriangledown^2 I_\lambda\right\}_{jk} = \frac{\partial^2 I_\lambda}{\partial c_j \partial c_k} \;=\; X'W_c X + n\lambda\Sigma, \tag{3.18}$$

where X is the matrix with ijth entry $B_j(t(i))$, p_c is the vector with ith entry $p_c(t(i))$ given by $p_c(t(i)) = \frac{e^{f_c(t(i))}}{(1+e^{f_c(t(i))})}$ where $f_c(\cdot) = \sum_{k=1}^{N} c_k B_k(\cdot)$, and W_c is the diagonal matrix with iith entry $p_c(t(i))(1-p_c(t(i)))$, compare (3.13, 3.14). We next describe the Newton-Raphson iterate for c. Given the ℓth Newton-Raphson iterate $c^{(\ell)}$, a straightforward calculation shows that $c^{(\ell+1)}$ is given by

$$c^{(\ell+1)} = c^{(\ell)} - (X'W_{c^{(\ell)}}X + n\lambda\Sigma)^{-1}(X'(y - p_{c^{(\ell)}}) + n\lambda\Sigma c^{(\ell)}) \tag{3.19}$$

and another straightforward calculation shows that $c^{(\ell+1)}$ is the minimizer of

$$I_\lambda^{(\ell)}(c) = ||z^{(\ell)} - W_{c(\ell)}^{1/2}Xc||^2 + n\lambda c'\Sigma c. \tag{3.20}$$

where $z^{(\ell)}$, the pseudo-data, is given by

$$z^{(\ell)} = W_{c(\ell)}^{-1/2}(y - p_{c(\ell)}) + W_{c(\ell)}^{1/2}Xc^{(\ell)}. \tag{3.21}$$

Next, we note that the 'predicted' value $\hat{z}^{(\ell)} = W_{c(\ell)}^{1/2}Xc$, where c is the minimizer of (3.20), is related to the pseudo-data $z^{(\ell)}$ by

$$\hat{z}^{(\ell)} = A^{(\ell)}(\lambda)z^{(\ell)}, \tag{3.22}$$

where $A^{(\ell)}(\lambda)$ is the smoother matrix given by

$$A^{(\ell)}(\lambda) = W_{c(\ell)}^{1/2}X(X'W_{c(\ell)}X + n\lambda\Sigma)^{-1}X'W_{c(\ell)}^{1/2} \tag{3.23}$$

In Wahba(1990), Section 9.2, [1] it was proposed to obtain a GCV score for λ in (3.16) as follows: For fixed λ, iterate (3.19) to convergence. Define $V^{(\ell)}(\lambda)$ as

$$V^{(\ell)}(\lambda) = \frac{\frac{1}{n}||(I - A^{(\ell)}(\lambda))z^{(\ell)}||^2}{\left(\frac{1}{n}tr(I - A^{(\ell)}(\lambda))\right)^2}. \tag{3.24}$$

Letting L be the converged value of ℓ, compute

$$V^{(L)}(\lambda) = \frac{\frac{1}{n}||(I - A^{(L)}(\lambda))z^{(L)}||^2}{\left(\frac{1}{n}tr(I - A^{(L)}(\lambda))\right)^2} \sim \frac{\frac{1}{n}||W_{c(L)}^{-1/2}(y - p_{c(L)})||^2}{\left(\frac{1}{n}tr(I - A^{(L)}(\lambda))\right)^2} \tag{3.25}$$

and minimize $V^{(L)}$ with respect to λ. Gu(1992b) found that the following algorithm for a GCV score for λ in this case was superior. Gu's algorithm goes as follows: Given a starting guess, from $c^{(\ell)}$, obtain $A^{(\ell)}(\lambda)$ and find $\lambda = \hat{\lambda}^{(\ell)}$ to minimize $V^{(\ell)}(\lambda)$; obtain $c^{(\ell+1)}$ by setting $\lambda = \hat{\lambda}^{(\ell)}$ in (3.19); iterate until convergence. We remark that the algorithms in Gu(1992b) and in Wahba(1990), and other algorithms, can be directly compared on simulated data by postulating a (synthetic) $p(\cdot)$ as 'truth', generating attribute vectors $t(i), i = 1, ..., n$, and generating the y_i for these attribute vectors by a random mechanism which lets y_i be 1 with probability $p(t(i))$. One can then estimate p by various methods. Since the 'true' p is known, an objective comparison can be made between

[1] The definition of λ there differs from the definition here by a factor of $n/2$. Please note the typographical error in (9.2.18) there where λ should be 2λ.

the 'true' p and the estimates, by computing the KL distance or other objective criterion. Gu made the comparison using the symmetrized KL distance ($= KL(p,\hat{p}) + KL(\hat{p},p)$).

Now, considering the numerator in the right hand side of (3.25), if we replace $||W_{c(\ell)}^{-1/2}(y-p_{c(\ell)})||^2$ by $||W^{-1/2}(y-p_{c(\ell)})||^2$ where W is the diagonal matrix with iith entry $p(t(i))(1-p(t(i)))$ we have a sum of squares of random variables involving $y_i/\sqrt{p(t(i))(1-p(t(i)))}$ with variance $\sigma^2 = 1$. This suggests replacing the approximate GCV estimate V of (3.25) with the UBR estimate

$$U^{(\ell)}(\lambda) = \frac{1}{n}||(I - A^{(\ell)}(\lambda))z^{(\ell)}||^2 + 2\frac{\sigma^2}{n}trA^{(\ell)}(\lambda) \tag{3.26}$$

with $\sigma^2 = 1$. Gu(1992b) suggests using this unbiased risk estimate computed via the following algorithm: Given a starting guess, from $c^{(\ell)}$, obtain $A^{(\ell)}(\lambda)$ and find $\lambda = \hat{\lambda}^{(\ell)}$ to minimize $U^{(\ell)}(\lambda)$; obtain $c^{(\ell+1)}$ by setting $\lambda = \hat{\lambda}^{(\ell)}$ in (3.19); iterate until convergence. Monte Carlo studies in Gu(1992b) suggest that this estimate is better than the approximate GCV estimate computed in a similar manner, based on a comparison of the symmetrized KL distance. In the remainder of this paper we will be using Gu's algorithm for the unbiased risk estimate for λ. It would be nice to have a good understanding of the difference between the 'iterate-to-convergence' algorithm and Gu's algorithm, both in the case of UBR and GCV. An argument in the UBR case is a little bit more transparent. To get a good estimate of λ from UBR in the Gaussian case it is clear that it is necessary to have a reasonably good estimate of σ, furthermore, two λ's compared via $U(\lambda)$ on the basis of different values of σ^2 cannot be expected to be comparable. In the UBR estimate here, the variances for different y_i are in general different, but if $p(t(i))(1-p(t(i)))$ were known, then the data would be rescaled by $\sqrt{p(t(i))(1-p(t(i)))}$ so that the variances of the rescaled data would all be 1. $\sqrt{p(t(i))(1-p(t(i)))}$ is not known, but the rescaling is being done implicitly with an estimate of it. If the iteration is carried to convergence before $U(\lambda)$ is minimized, then the rescaling is being done with different estimates of the standard deviations for different λ's and the comparison of different λ's by looking at $U(\lambda)$ is not necessarily valid. In Gu's algorithm, at the ℓ'th iteration different λ's are being compared based on the *same* estimate $diag\ W_{c(\ell)}$ of the variances of the y_i, so, at each iteration at least, $U(\lambda)$ for different $\lambda's$ can be expected to be more directly comparable. O'Sullivan(1988) has considered approximate UBR estimates in the case of penalized log-density and log-hazard estimates which involve the computation of an estimated degrees of freedom for signal. However the estimate there does not include an implicit estimate of a variable variance.

Moody(1991) discusses what may be considered a generalization of the unbiased risk estimate in situations which are in one sense more general than the setup we have been discussing. Since his generalization raises an important question for the case when the penalty functional or regularizer is not quadratic, we will discuss the relationship of his estimate to the UBR discussed here and note the issue. Moody assumes (in our notation) the model (3.1), where he assumes that the ϵ_i are independently distributed with mean 0 and common variance σ^2, (either known or estimated) but not necessarily Gaussian. He considers more general methods of approximating f, he mentions 'multilayer perceptrons and radial basis functions or other learning systems'. Once the architecture is determined, he says that f will be estimated by f_ω determined by a set of weights ω_i (which play the role of our c), which in turn depend on a smoothing parameter λ. Moody next defines $\mathcal{E}(y,\omega)$ by

$$\mathcal{E}(y,\omega) = \frac{1}{n}\sum_{i=1}^{n} E(y_i, f_\omega(t(i))) \tag{3.27}$$

where $E(y_i, f_\omega(t(i))$ is an unspecified distance between y_i and $f_\omega(t(i)))$. For fixed λ, $f = f_{\lambda,\omega}$, is obtained by finding ω to minimize

$$I_\lambda(y,\omega) = \mathcal{E}(y,\omega) + \lambda S(\omega) \tag{3.28}$$

where $S(\omega)$ is a 'smoothness' penalty on f_ω, not necessarily quadratic. Moody proposes that λ be estimated as the minimizer of $\mathcal{E}(y,w) + 2\frac{\sigma^2}{n}\rho_{eff}(\lambda)$, where $\rho_{eff}(\lambda)$ is called by Moody 'the effective number of parameters' and by us the 'degrees of freedom for signal', and is the trace of what he calls the generalized influence matrix $G = \frac{1}{2}\tilde{X}H^{-1}\tilde{X}'$ where $\tilde{X}$ and H are the matrices defined by

$$\tilde{X}_{i\alpha} = \frac{\partial}{\partial y_i}\frac{\partial}{\partial \omega_\alpha}\mathcal{E}(\omega) \tag{3.29}$$

$$H_{\alpha\beta} = \frac{\partial}{\partial \omega_\alpha}\frac{\partial}{\partial \omega_\beta} I_\lambda(\omega). \tag{3.30}$$

It is easy to check that G plays the role of the influence matrix $A(\lambda)$ in the case that $\mathcal{E}$ and S are both quadratic with the y_i's treated as though they were independent with a common known variance, by considering

$$I_\lambda(y,\omega) = \|y - X\omega\|^2 + n\lambda\omega'\Sigma\omega, \tag{3.31}$$

then $\tilde{X} = 2X, H = 2(X'X + n\lambda\Sigma)$, and $G = X(X'X + n\lambda\Sigma)^{-1}X'$, compare (3.20), (3.22) and (3.23). If, however, $\frac{\partial}{\partial y_i}\frac{\partial}{\partial \omega_\alpha}\mathcal{E}(y,\omega)$ or $\frac{\partial}{\partial \omega_\alpha}\frac{\partial}{\partial \omega_\beta}S(\omega)$ depends on ω, then it is possible that *where* the derivatives are taken will make a difference, just as happens in the case considered earlier with $\mathcal{E}$ taken as $\mathcal{L}$.

We remark that in an entirely different context, varying df-signal can be related to varying the stopping criterion in an iterative fitting method, see Wahba(1987), we suspect that this phenomenon is fairly general in NN algorithms.

4 Smoothing spline analysis of variance (SS-ANOVA)

In the ANOVA approach to estimating a function of d variables, $f(t) = f(t_1, ..., t_d)$ is decomposed as

$$f(t) = \mu + \sum_{\alpha} f_\alpha(t_\alpha) + \sum_{\alpha<\beta} f_{\alpha\beta}(t_\alpha, t_\beta) + \sum_{\alpha<\beta<\gamma} f_{\alpha\beta\gamma}(t_\alpha, t_\beta, t_\gamma) + \cdots \tag{4.1}$$

where the elements in the expansion are made unique in some manner or other, and, the expansion is truncated in some manner. See, for example Stone(1985), Friedman(1991), Buja, Hastie and Tibshirani(1989). In the smoothing spline ANOVA context, with Gaussian data, the estimate $f_{\lambda,\theta}$ of f is obtained by finding $f_{\lambda,\theta}$ of the form of (4.1) in an appropriate function space (a reproducing kernel Hilbert space) to minimize

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(t(i)))^2 + \lambda J_\theta(f) \tag{4.2}$$

where

$$J_\theta(f) = \sum_{\alpha\in\mathcal{M}} \theta_\alpha^{-1} J_\alpha(f_\alpha) + \sum_{\alpha,\beta\in\mathcal{M}} \theta_{\alpha\beta}^{-1} J_{\alpha\beta}(f_{\alpha\beta}) + \cdots \tag{4.3}$$

The referenced function space has been constructed so that the mean μ, the main effects f_α, the two factor interactions $f_{\alpha\beta}$ and so forth are projections onto orthogonal subspaces whose elements satisfy certain side conditions. This generalizes the usual ANOVA decomposition familiar to veterans of some introductory statistics courses, see for example Hogg and Ledolter(1987), Chapter 6. Here $\mathcal{M}$ is the collection of indices for components with penalty functionals to be included in the model, the $J_\alpha, J_{\alpha\beta}$ and so forth are quadratic 'smoothness' penalty functionals, and λ and the θ_β's satisfy an appropriate constraint for identifiability.

We will first describe what happens in this quadratic (Gaussian) context, then we will show how code in the quadratic case can be used as a subroutine in the computation of a PSA model for soft classification. References for the quadratic (SS-ANOVA) case are Gu(1989), Gu, Bates, Chen and Wahba(1989), Chen, Gu and Wahba(1989), Gu and Wahba(1991a,b,1993a,b), Wahba(1986,1990).

First, for convenience, linearly relabel the, say, q terms included in the model in (4.3) so that β may stand for $\alpha, \alpha\beta, \alpha\beta\gamma$ and so forth, to obtain

$$J_\theta(f) = \sum_{\beta=1}^{q} \theta_\beta^{-1} J_\beta(f_\beta). \tag{4.4}$$

It is known in the SS-ANOVA setup that the minimizer of (4.2) is in the n dimensional space of functions with representation

$$f_{\lambda,\theta}(t) = \sum_{\nu=1}^{M} d_\nu \phi_\nu(t) + \sum_{i=1}^{n} c_i \sum_{\beta=1}^{q} \theta_\beta R_\beta(t, t(i)) \tag{4.5}$$

where the ϕ_ν span the null space of the penalty functional J_θ, the $R_\beta(\cdot,\cdot)$ are certain reproducing kernels associated with the corresponding terms J_β in the penalty functional and the $\{c_i\}$ satisfy the M conditions $\sum_{i=1}^n c_i\phi_\nu(t(i)) = 0, \nu = 1, ..., M$. Letting Σ_θ be the $n \times n$ matrix with ijth entry $\sum_{\beta=1}^q \theta_\beta R_\beta(t(i), t(j))$, the coefficients $d = (d_1, ..., d_M)'$ and $c = (c_1, ..., c_n)'$, are obtained by substituting (4.5) into (4.2) which then becomes [2]

$$\frac{1}{n}||y - (\Sigma_\theta c + Td)||^2 + \lambda c'\Sigma_\theta c. \tag{4.6}$$

The minimizing (c, d) satisfy

$$\begin{aligned} (\Sigma_\theta + n\lambda I)c + Td &= y \qquad (4.7) \\ T'c &= 0. \qquad (4.8) \end{aligned}$$

See Wahba(1990), Chapter 10. The generic code RKPACK(Gu(1989)) can be used to compute $trA(\lambda,\theta)$, where $A(\lambda,\theta)$ is the matrix which satisfies $\hat{y} = A(\lambda,\theta)y$, where $\hat{y} = (\Sigma_\theta c + Td)$, to determine $\lambda/\theta_\beta, \beta = 1, 2, .., q$, by UBR or GCV, and to obtain c and d, given the ingredients y, Σ_θ and T, and in the case of UBR, σ^2.

We now return to the problem of soft classification, where we suppose that y_i is 1 or 0 with probability $p(t(i))$, and where $t = (t_1, ..., d_d)$. We suppose that $f(t) = log[p(t)/(1 - p(t))]$ as before, but we will model f as

$$f(t) = \mu + \sum_{\alpha\in\mathcal{M}} f_\alpha(t_\alpha) + \sum_{\alpha,\beta\in\mathcal{M}} f_{\alpha\beta}(t_\alpha, t_\beta) + \cdots \tag{4.9}$$

[2] By the properties of reproducing kernels it can be shown in the setup discussed here that $J_\theta(\sum c_i \sum \theta_\beta R_\beta(\cdot, t(i)) = c'\Sigma_\theta c$.

Replacing (4.2) by

$$\mathcal{L}(y,f) + \frac{n}{2}\lambda J_\theta(f) \tag{4.10}$$

where $\mathcal{L}(y,f)$ is given by (2.4), it can be shown that the minimizer has a representation (4.5) for some (c,d), with the same M conditions on c. Furthermore, it is shown in Gu(1990) [3] that the Newton iterate $(c^{(\ell+1)}, d^{(\ell+1)})$ for the minimization of (4.10) with f as in (4.5), is the minimizer of

$$I^{(\ell)}_{\lambda,\theta}(c,d) = \|z^{(\ell)} - W^{1/2}_{(\ell)}(\Sigma_\theta c + Td)\|^2 + n\lambda c'\Sigma_\theta c, \tag{4.11}$$

where we are here and below writing the subscript (ℓ) as shorthand for the subscript $(c^{(\ell)}, d^{(\ell)})$, and $z^{(\ell)}$, the pseudo-data, is given by

$$z^{(\ell)} = W^{-1/2}_{(\ell)}(y - p_{(\ell)}) + W^{1/2}_{(\ell)}(\Sigma_\theta c^{(\ell)} + Td^{(\ell)}), \tag{4.12}$$

compare (3.20,3.21). For fixed (ℓ), make the change of variables in (4.11):

$$\begin{aligned}
\tilde{c} &= W^{-1/2}_{(\ell)} c, && (4.13)\\
\tilde{\Sigma}_\theta &= W^{1/2}_{(\ell)} \Sigma_\theta W^{1/2}_{(\ell)}, && (4.14)\\
\tilde{T} &= W^{1/2}_{(\ell)} T && (4.15)\\
\tilde{d} &= d && (4.16)
\end{aligned}$$

to obtain

$$I^{(\ell)}_{\lambda,\theta}(\tilde{c},\tilde{d}) = \|z^{(\ell)} - (\tilde{\Sigma}_\theta\tilde{c} + \tilde{T}\tilde{d})\|^2 + n\lambda\tilde{c}'\tilde{\Sigma}_\theta\tilde{c}. \tag{4.17}$$

Equation (4.17) is of the same form as (4.6). $A^{(\ell)}(\lambda,\theta)$ is the matrix which satisfies $\hat{z}^{(\ell)} = A^{(\ell)}(\lambda,\theta)z^{(\ell)}$, where $\hat{z}^{(\ell)} = (\tilde{\Sigma}_\theta\tilde{c} + \tilde{T}\tilde{d})$. Given the ingredients $z^{(\ell)}, \tilde{\Sigma}_\theta$, and $\tilde{T}$, RKPACK can be called at the (ℓ)th step as a subroutine to obtain the UBR (or GCV) estimates of λ and θ_β, and (then) the Newton update with these updated values of the smoothing parameters. (RKPACK imposes conditions guaranteeing uniqueness, the solution only depends on the ratios λ/θ_β). Thus, the algorithm is: given a starting guess, from $c^{(\ell)}, d^{(\ell)}$, obtain $A^{(\ell)}(\lambda,\theta)$ and find $\lambda, \theta_\beta = \hat{\lambda}^{(\ell)}, \hat{\theta}^{(\ell)}_\beta$ to minimize $U^{(\ell)}(\lambda,\theta)$ given by (3.9) with $\sigma^2 = 1$, and $A^{(\ell)}(\lambda)$ replaced by $A^{(\ell)}(\lambda,\theta)$; obtain $c^{(\ell+1)}, d^{(\ell+1)}$ from (4.11) with $\lambda,\theta = \hat{\lambda}^{(\ell)}, \hat{\theta}^{(\ell)}$; iterate until convergence.

[3] Please note the following typographical errors in Gu(1990): $\tilde{c} = W_-^{-1/2}c$ and not $W_-^{1/2}c$, $\tilde{y} = W_-^{-1/2}(W_-\eta_- - u_-)$ and not $\tilde{y} = W_-^{1/2}(W_-\eta_- - u_-)$, also w_j in (2.6) should be w_{j-}

5 Application to the Pima-Indian data set

We have built and compared several PSA models for estimating $p(t)$ on the Pima Indians Diabetes Database which we retrieved from the UCI Repository of Machine Learning Databases and Domain Theories (ics.uci.edu: pub/machine-learning-databases) on October 7, 1992. This data set has also been analyzed using the ADAP learning algorithm by Smith *et al*(1988) so we will be able to compare some of our results with theirs. This database contains records of 768 instances, which were medical records from Pima-Indian women at least 21 years of age. Below is reproduced the list of 8 attribute variables and the class variable (response):

```
1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)
```

The class variable was an indicator (1) for a positive test for diabetes between 1 and 5 years from the examination determining the other variables, or (0) a negative test for diabetes 5 or more years later. The repository index reports that there were 268 cases with '1' as their indicator and 500 with '0'. It also reports that there are no missing attribute values, however, after some investigation into peculiar behavior of some of our results, box-plots of each set of attribute values revealed that there were 11 instances of 0 body mass index and 5 instances of 0 plasma glucose, both physical impossibilities(!). We have deleted those cases, leaving 752 instances for our experiments. Smith *et al*(1988) report that they used 576 randomly selected cases to train the ADAP algorithm, and then used the remaining 192 test cases as an evaluation set, to study the properties of the trained ADAP algorithm. Smith *et al*(1988) note that the ADAP algorithm is an 'interactive associative learning model using the Hebbian learning rule', and give a brief description of the algorithm and further references. Given a new instance, the algorithm will output a score (real number) which is evidently intended to be larger if the new instance is more like the training 1's and smaller if the new instance is more like the training 0's. However, there is no suggestion that this score is intended to have the meaning of a probability. In any case, once a threshold level on this score is chosen, a 'hard' classification (forecast) is made. The authors present a plot of specificity and sensitivity against a horizontal scale consisting of the ranks of the individuals in the the evaluation set ordered with respect to the output score. The sensitivity as a function of the rank is the fraction of true positives in the evaluation set with higher rank, and the specificity is the fraction of true negatives in the evaluation set with smaller rank. Thus, one can read the false positive and false negative rates off this plot as a function of threshold rank (or score, if it were provided). By inspection of the curves it appears that they cross at about rank 112 (out of 192). Smith *et al*(1988) report that the score at the crossing point is .448, and if this score is used to enforce a 'hard' classification, then the rate of successful classification for both the true negatives and the true positives is 76%.

Our existing code is a big time and storage hog, as a result we found it necessary to be more modest than we would like in the data analysis. Thus, we decided to see how well we could do with fewer variables, and with a somewhat smaller training set. We randomly selected 500 instances out of 752 for the training set, and set aside the remaining 252 as the evaluation set. We used the `glm` function in S (Becker, Chambers and Wilks(1988)), which implements the GLIM models of McCullagh and Nelder(1989), to fit several parametric models to the data in an effort to select a few of the most influential predictors. The GLIM model finds $f(t_1, ..., t_d)$ as a linear combination of simple parametric functions in the variables $(t_1, ...t_d)$. The linear GLIM model would, for example set $f(t_1, ..., t_d) = a + \sum_{\alpha=1}^{d} b_\alpha t_\alpha$, and find a and the b_α to minimize $\mathcal{L}(y, f)$ over f's of this form. The linear GLIM fit suggested that variables 1,2,6 and 7 were 'significant' (assuming that you believed this model). Running one variable at a time through the linear GLIM model gave all relatively poor fits to the data compared to models with more than one variable, as measured by

$$\widehat{KL}_{EVAL} = \frac{1}{252} \sum_{i=1}^{252} [y_i \hat{f}(t(i)) - log(1 + e^{\hat{f}(t(i))})]. \tag{5.1}$$

where $\hat{f}$ is the GLIM model based on the training data and the sum in $\widehat{KL}_{EVAL}$ is over the evaluation data. Running the variables two at a time, the best pairwise variables according to $\widehat{KL}_{EVAL}$ were (2,6), (1,2) and (2,7), in that order, and the best of the three variable combinations was (1,2,6). For the application of PSA we decided to concentrate on a two variable model (2,6), a three variable model(1,2,6), and a four variable model (1,2,6,7). We considered variables 2, 6 and 7 as continuous variables, but we decided to consider variable 1 (number of pregnancies) as a categorical variable, with the four categories $C_1 = 0, C_2 = \{1,2\}, C_3 = \{3,4,5\}$ and $C_4 = \{> 5\}$ We considered the following four models: Model I: $f(t) = \mu + f_2(t_2) + f_6(t_6)$, Model II: $f(t) = \mu + f_2(t_2) + f_6(t_6) + f_{2,6}(t_2, t_6)$, Model III: $f(t) = \mu + \sum_{k=1}^{3} \gamma_k I_k(t_1) + f_2(t_2) + f_6(t_6) + f_{2,6}(t_2, t_6)$, where $I_k(t_k)$ is an indicator function which is 1 if variable 1 is from category C_k and 0 otherwise, and Model IV: $f(t) = \mu + \sum_{k=1}^{3} \gamma_k I_k(t_1) + f_2(t_2) + f_6(t_6) + f_7(t_7)$. Models III and IV, which have linear combinations of a small number of unpenalized functions of known form (here indicator functions), are known as partial spline models, see Wahba(1990). From an algorithmic point of view, these functions are simply added to the set of functions spanning the null space of the penalty functional. [4]

Each of these models have a GLIM model as a special case, which is obtained by replacing $\mu + f_2(t_2) + f_6(t_6)$ by $\mu + a_2 t_2 + a_6 t_6$ or $\mu + f_2(t_2) + f_6(t_6) + f_{2,6}(t_2, t_6)$ by $\mu + a_2 t_2 + a_6 t_6 + a_{2,6} t_2 t_6$, and similarly for t_7. The GLIM model would be fitted as a special case of the corresponding PSA model if all the λ/θ_β were estimated as ∞.

We can now compare all eight these models by looking at their action on the 252 cases that have been left out. We estimate (2.9) for the PSA models by

$$\widehat{KL}_{EVAL} = \frac{1}{252} \sum_{i=1}^{252} [y_i f_{\widehat{\lambda,\theta}}(t(i)) - \log(1 + e^{f_{\widehat{\lambda,\theta}}(t(i))})], \tag{5.2}$$

where again the sum is over the evaluation data and $f_{\widehat{\lambda,\theta}}$ is one of the four models that has been fit on the training set, using Gu's algorithm for the UBR to get $\widehat{\lambda,\theta}$. See Seaman and Hutchinson(1985), Wolpert(1992), and Schaffer(1993) for closely related approaches to this *model selection* problem, and Gu(1992a) and Gu and Wahba(1993b) for philosophically different approaches to this problem, based on the sizes of each estimated component rather than a predictive criterion as we are using here. It would of course no longer be 'fair' to compare the best of these models against the ADAP or other model on the basis of the *same* evaluation set, since the evaluation set has now been used to select the model. A 'fair' comparison would be to take the model selected this way and compare

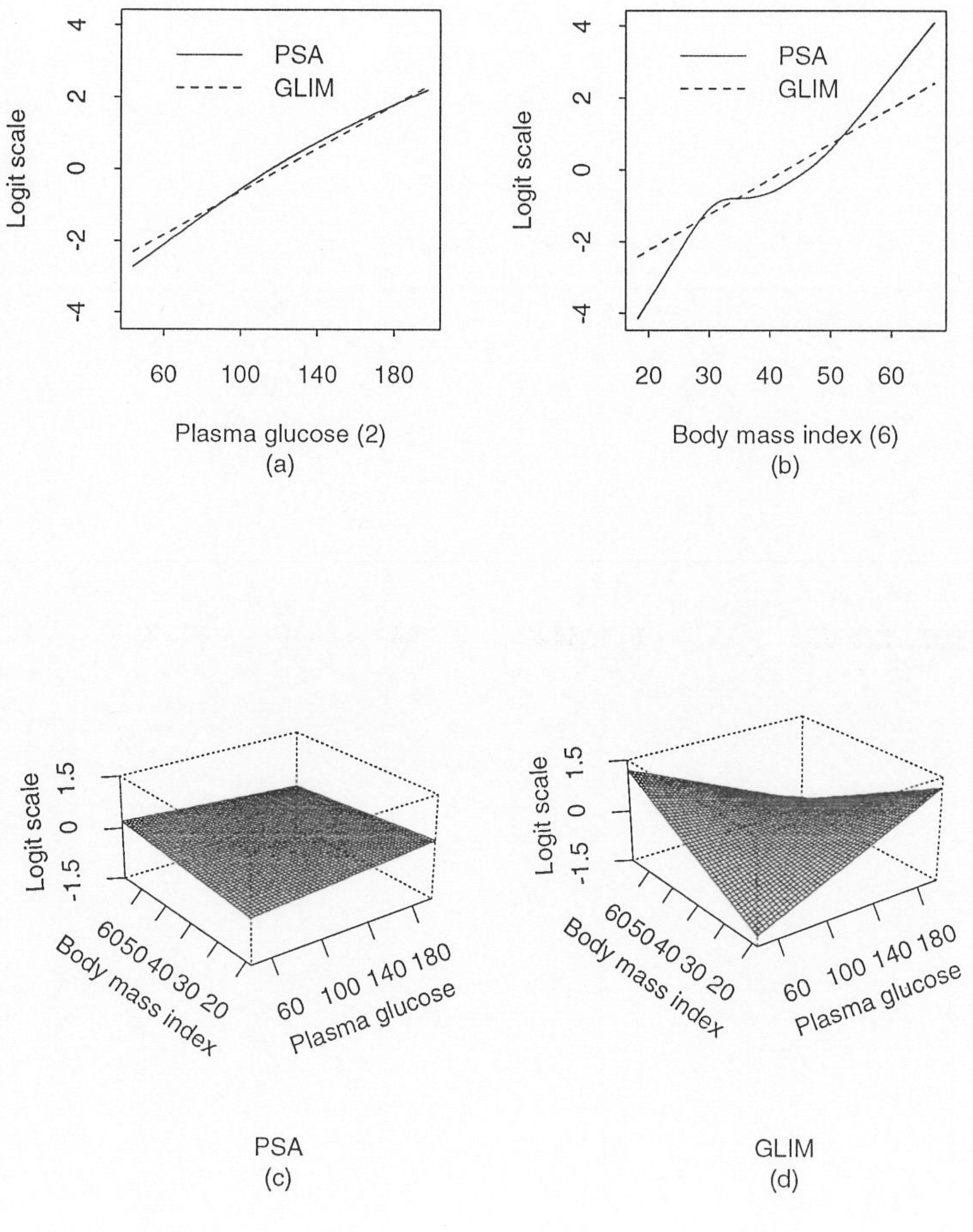
[4] The representations (4.5) for these models were constructed as in Gu and Wahba(1993b) by rescaling t_2, t_6 and t_7 to the unit cube with the largest and smallest values mapped to 1 and 0, and using the reproducing kernels of (4.1) and (4.2) of that paper as building blocks. The main effects penalty functionals were $J_\alpha(f_\alpha) = \int_0^1 (f''(t_\alpha))^2 dt_\alpha$, see Gu and Wahba(1993a,b) for further details. Gu and Wahba(1993b) is available until publication from `wahba@stat.wisc.edu` as UW-Madison Statistics Dept. TR 881.

it against a competitor on *another* evaluation set. Table 5.1 gives $\widehat{KL}_{EVAL}$ the four PSA models and the four corresponding GLIM models.

Model	PSA	GLIM
I	0.4929	0.5075
II	0.4861	0.5004
III	0.3925	0.5222
IV	0.4157	0.5226

Table 5.1: $\widehat{KL}_{EVAL}$

The table identifies the PSA Model III as the 'winner'. We do not (as yet) have an objective criterion for saying which, if any of these 8 models are significantly different (either in a statistical or a practical sense) from the 'winner' although it appears that theoretical criteria relating to statistical significance can be developed. We remark that in PSA Model IV the main effect for variable 7, diabetes pedigree function, was quite small compared to the other main effects, and so PSA models III and IV were very similar, in a practical sense. We will restrict further description to PSA Model III and its GLIM counterpart. The solid lines in Figure 5.1 (a) and (b) give the main effects f_2 and f_6 for variables 2 (plasma glucose) and 6 (body mass index) for PSA Model III, and Figure 5.1 (c) gives the interaction term for PSA Model III. The dashed lines in Figure 5.1 (a) and (b) give, for comparison, the corresponding GLIM main effects, which are, by construction, straight lines. and Figure 5.1 (d) gives the GLIM interaction term, which is of necessity bilinear in t_2 and t_6. The fitted interaction term in the PSA model actually was estimated as very close to bilinear in t_2 and t_6, and very small, possibly negligible. The interaction term in the GLIM model, which does not have a ready interpretation, was not statistically significant, according to the criteria of the GLIM code, which assumes that some GLIM model is true. Figure 5.2 gives contour plots of $\hat{p}(t_1, t_2, t_6)$ corresponding to $t_1 = \{0\}, \{1,2\}, \{3,4\}$ and $\{> 5\}$ pregnancies, as a function of variables 2 and 6. Visually the plots for the first three categories do not appear much different, but being in the fourth category (≥ 5 pregnancies) appears to increase the risk at all levels of (t_2, t_6). On a logit scale, the difference between the fourth category and the average of the first three was about .92. Figure 5.3 gives the same contour plots based on the GLIM model.



(a)

(b)

PSA
(c)

GLIM
(d)

Figure 5.1: Logit Main Effects and Interaction for PSA Model III and GLIM Model III

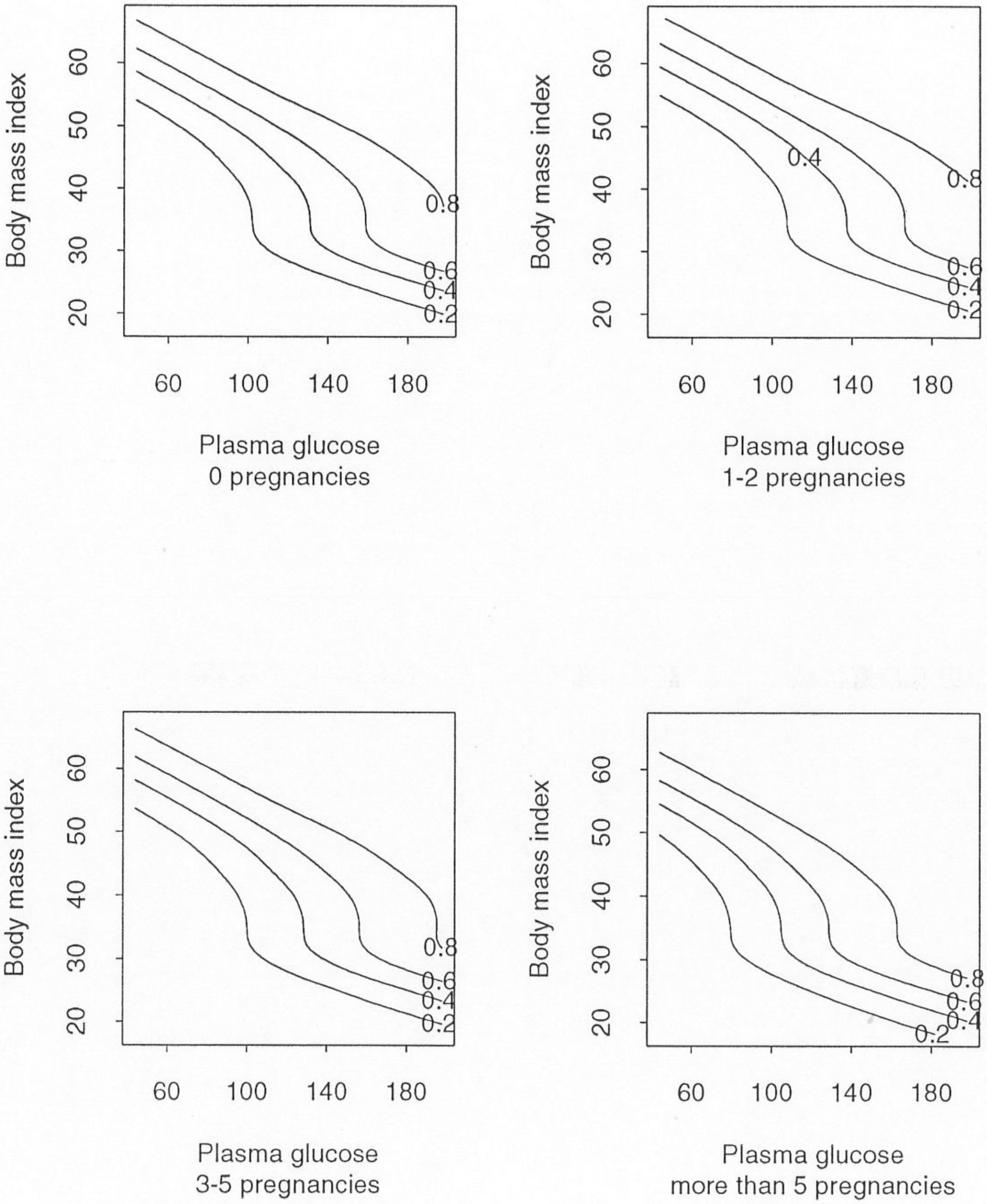


Figure 5.2: Probability Estimates from the PSA Model III, for the Four Categories of Variable 1, as a Function of Variables 2 and 6.

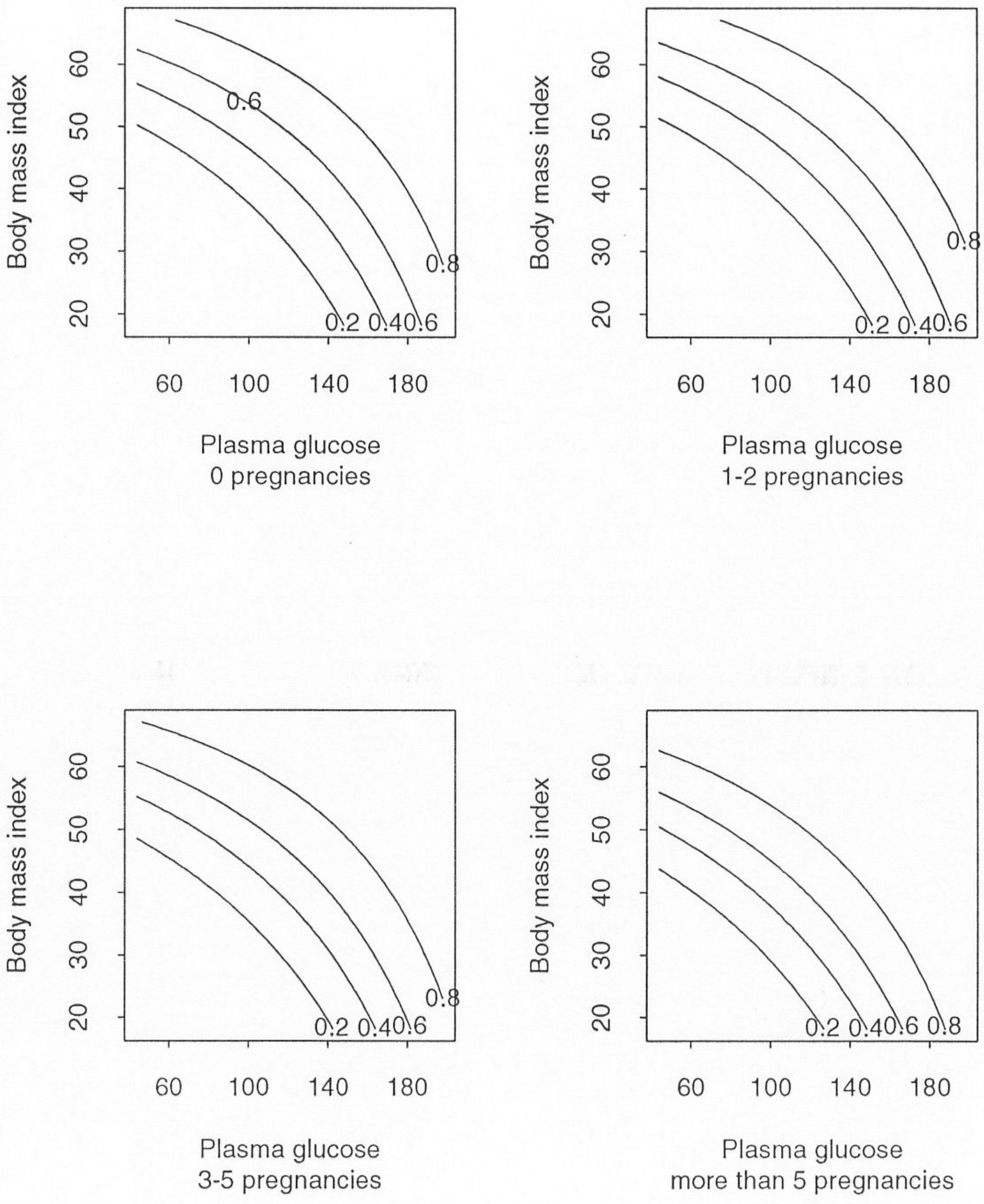


Figure 5.3: Probability Estimates from the GLIM Model III, for the Four Categories of Variable 1, as a Function of Variables 2 and 6.

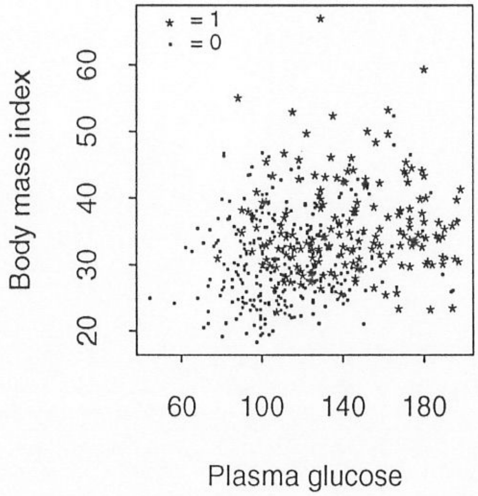


Figure 5.4: Scatterplot of Body Mass Index and Plasma Glucose.

Figure 5.4 gives a plot of the body mass index *vs* plasma glucose for the 500 member training set, 1's are plotted with a '*' and 0's are plotted with a '·'. There are just two cases of body mass index above 55, so the models should not be taken too seriously much past 55. Methods for providing confidence statements for these model outputs which suggest the regions in which they can be trusted are discussed in Gu(1992c) and Gu and Wahba(1993b).

Note that in this population all of the cases with body mass index less than about 23 did not later turn out to have diabetes. Similarly all of the cases with plasma glucose less than 78 did not later turn out to have diabetes. The greater flexibility in the PSA model allows the main effects to drop much more steeply than the GLIM model to accommodate this. Note also that the contribution of body mass index main effects in the PSA model is fairly flat along the range of about 30 to 40 body mass index, suggesting the desirability of keeping one's body mass index in that range.

In order to compare this method with the analysis in Smith *et al*(1988) we give in Figure 5.5(a) a plot of the sensitivity and the specificity scores for PSA and GLIM Model III, plotted against the ranked evaluation data, ranked according to the probability estimate. The sensitivity and specificity curves cross at about rank 146 (out of 252). If the score at this rank (which was about

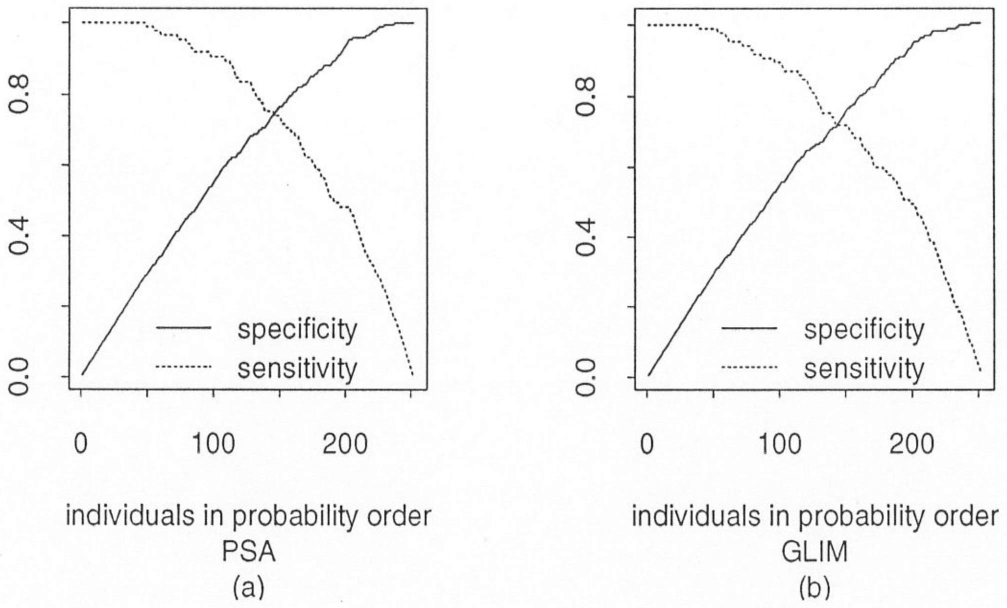
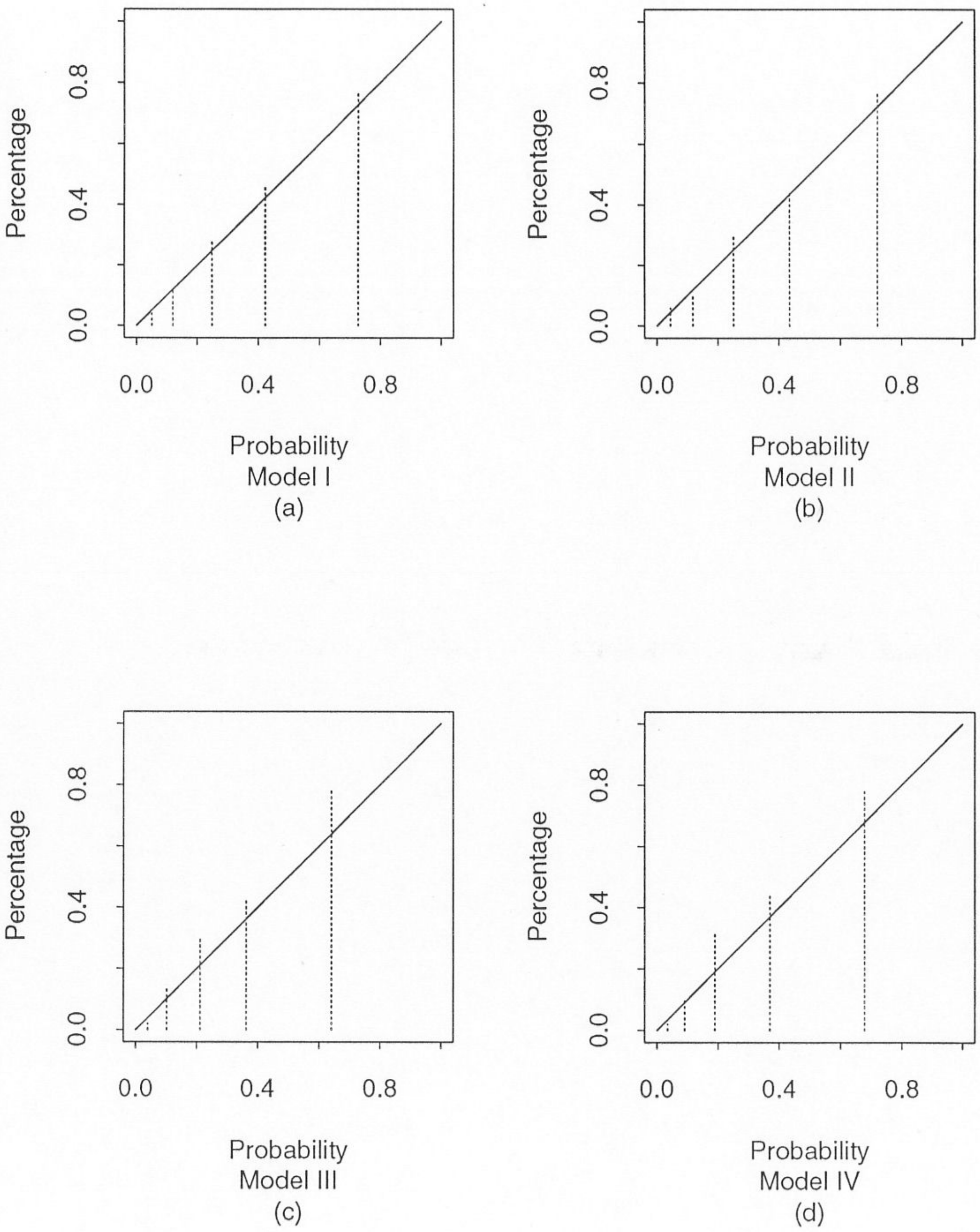


Figure 5.5: Specificity and Sensitivity for the PSA Model III and GLIM Model III.

.74) , were used to enforce a 'hard' classification, then the rate of successful classification for both the true negatives and the true positives in the evaluation sample would be about 74%. Figure 5.5 (b) gives the same plots for the GLIM Model III. The successful classification rate at the crossover point for GLIM Model III was about 72%. The sensitivity *vs* specificity curve given in Smith *et al*(1988) is visually very similar to Figure 5.5(a), the corresponding success rate for the ADAP analysis, using the crossover point as threshold was reported as 76%. This raises an interesting point about the ADAP algorithm. Its score report is evidently not claimed to be a probability, however, the Neyman-Pearson Lemma tells us that if we knew the correct probability, then any optimum threshold would depend on it, thus, we might ask if the ADAP score (or other neural net scores) can be reasonably transformed into a probability by a monotone transformation. For more on this point, see Richard and Lippman(1991). In Figure 5.6, we provide a plot that suggests that the 'score' $\hat{p}$ really does have a reasonable meaning as a probability. The evaluation data set of 252 cases has been rank ordered by their associated $\hat{p}$'s and then arbitrarily divided into 5 groups of size 50,50,50,50 and 52 respectively. Returning to our 'world view', suppose that the 252 probability



Model I
(a)

Model II
(b)

Model III
(c)

Model IV
(d)

Figure 5.6: Estimates of Observed Fraction of 1's *vs* Expected Fraction, by Quintile, PSA models.

estimates associated with these 252 cases represented 'true' probabilities - say the first 50 values were $p_1, ..., p_{50}$. Then the expected fraction of 1's in this group would be $\frac{1}{50}\sum_{k=1}^{50} p_k$. In Figure 5.6 (a) -(d) we have plotted the observed fractions for the five groups against their expected fractions, for the Spline Models I-IV. If the estimates were 'on the money', they would fall on the solid line. We were somewhat surprised that Model I appears visually to be more 'on the money' than Model III by this criterion, which of course is not exactly the same as the KL distance. At this point we have not made a study of the variability of these comparisons.

References

R. Becker, J. Chambers, and A. Wilks (1988). *The New S Language.* Wadsworth.

A. Buja, T. Hastie, and R. Tibshirani (1989). "Linear smoothers and additive models" (with discussion), *Ann. Statist.*, 17, 453 – 555.

W. Buntine and A. Weigend (1991). "Bayesian back-propagation," *Complex systems*, 5, 603 – 643.

Z. Chen, C. Gu, and G. Wahba (1989). Comments to "Linear Smoothers and Additive Models" by Buja, Hastie and Tibshirani, *Ann. Statist.*, 17, 515 – 521.

D. Cox and Y. Chang (1990). Iterated state space algorithms and cross validation for generalized smoothing splines. Technical Report 49, University of Illinois, Dept. of Statistics.

P. Craven and G. Wahba (1979). "Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation," *Numer. Math.*, 31, 377 – 403.

J. Friedman (1991). "Multivariate adaptive regression splines" (with discussion), *Ann. Statist*, 19, 1 – 141.

S. Geman, E. Bienenstock, and R. Doursat (1992). "Neural networks and the bias/variance dilemma," *Neural Computation*, 4, 1 – 58.

R. Gray (1992). "Flexible methods for analyzing survival data using splines with applications to breast cancer prognosis," *J. Amer. Statist. Assoc.*, 87, 942 – 951.

C. Gu (1989). "RKPACK and its applications: Fitting smoothing spline models," In *Proceedings of the Statistical Computing Section*, pp. 42 – 51. American Statistical Association.

C. Gu (1990). "Adaptive spline smoothing in non-Gaussian regression models," *J. Amer. Statist. Assoc.*, 85, 801 – 807.

C. Gu (1992a). "Diagnostics for nonparametric regression models with additive terms," *J. Amer. Statist. Assoc.*, 87, 1051 – 1057.

C. Gu (1992b). "Cross-validating non-Gaussian data," *J. Comput. Graph. Statist.*, 1, 169 – 179.

C. Gu (1992c). "Penalized likelihood regression: A Bayesian analysis," *Statist. Sin.*, 2, 255 – 264.

C. Gu, D.M. Bates, Z. Chen, and G. Wahba (1989). "The computation of GCV functions through Householder tridiagonalization with application to the fitting of interaction spline models," *SIAM J. Matrix Anal.*, 10, 457 – 480.

C. Gu and G. Wahba (1991a). Comments to "Multivariate adaptive regression splines" by Friedman, *Ann. Statist.*, 19, 115 – 123.

C. Gu and G. Wahba (1991b). "Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method," *SIAM J. Sci. Statist. Comput.*, 12, 383 – 398.

C. Gu and G. Wahba (1993a). "Semiparametric ANOVA with tensor product thin plate splines," *J. Roy. Statist. Soc. Ser. B*, 55, to appear.

C. Gu and G. Wahba (1993b). "Smoothing spline ANOVA with component-wise Bayesian 'confidence intervals'," *J. Comput. Graph. Statist.*, 2, to appear.

R. Hogg and J. Ledolter (1987). *Engineering Statistics.* Macmillan.

G. Kimeldorf and G. Wahba (1970). "A correspondence between Bayesian estimation of stochastic processes and smoothing by splines," *Ann. Math. Statist.*, 41, 495 – 502.

G. Kimeldorf and G. Wahba (1971). "Some results on Tchebycheffian spline functions," *J. Math. Anal. Applic.*, 33, 82 – 95.

K. C. Li (1985). "From Stein's unbiased risk estimates to the method of generalized cross-validation," *Ann. Statist.*, 13, 1352 – 1377.

K. C. Li (1986). "Asymptotic optimality of C_L and generalized cross validation in ridge regression with application to spline smoothing," *Ann. Statist.*, 14, 1101 – 1112.

M. Richard and R. Lippmann (1991). "Neural network classifiers estimate Bayesian *a posteriori* probabilities," *Neural Computation*, 3, 461 – 483.

C. Mallows (1973). "Some comments on C_p," *Technometrics*, 15, 661 – 675.

P. McCullagh and J. Nelder (1989). *Generalized Linear Models (2nd ed.).* Chapman and Hall.

J. Moody (1991). "The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems," In J. Moody, S. Hanson, and R. Lippman, editors, *Advances in Neural Information Processing Systems 4.* Kaufmann.

F. O'Sullivan (1983). *The analysis of some penalized likelihood estimation schemes.* PhD thesis, University of Wisconsin, Dept. of Statistics.

F. O'Sullivan (1988). "Fast computation of fully automated log-density and log-hazard estimators," *SIAM J. Sci. Statist. Comput.*, 9, 363 – 379.

F. O'Sullivan, B. Yandell, and W. Raynor (1986). "Automatic smoothing of regression functions in generalized linear models," *J. Amer. Statist. Assoc.*, 81, 96 – 103.

B. Ripley (1992). Neual networks and related methods for classification. manuscript, submitted to *J. Roy. Statist. Soc.*, available by anonymous ftp from `markov.stats.ox.ac.uk`.

C. Schaffer (1993). "Selecting a classification method by cross-validation," *Machine Learning*, to appear.

R. Seaman and M. Hutchinson (1985). "Comparative real data tests of some objective analysis methods by withholding," *Aust. Met. Mag.*, 33, 37 – 46.

J. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes (1988). "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," In *Proceedings of the Symposium on Computer Applications and Medical Care*, pp. 261 – 265. IEEE Computer Society Press.

C. Stone (1985). "Additive regression and other nonparametric models," *Ann. Statist.*, 13, 689 – 705.

R. Tibshirani and M. LeBlanc (1992). "A strategy for binary description and classification," *J. Comput. Graph Statist.*, 1, 3 – 20.

G. Wahba (1978). "Improper priors, spline smoothing and the problem of guarding against model errors in regression," *J. Roy. Statist. Soc. Ser. B*, 40, 364 – 372.

G. Wahba (1983). "Bayesian 'confidence intervals' for the cross-validated smoothing spline," *J. Roy. Statist. Soc. Ser. B*, 45, 133 – 150.

G. Wahba (1985). "A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem," *Ann. Statist.*, 13, 1378 – 1402.

G. Wahba (1986). "Partial and interaction splines for the semiparametric estimation of functions of several variables," In T. Boardman, editor, *Computer Science and Statistics: Proceedings of the 18th Symposium*, pp. 75 – 80. American Statistical Association.

G. Wahba (1987). "Three topics in ill posed problems," In H. Engl and C. Groetsch, editors, *Proceedings of the Alpine-U.S. Seminar on Inverse and Ill Posed Problems*, pp. 37 – 51. Academic Press.

G. Wahba (1990). *Spline Models for Observational Data.* CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 59. SIAM.

G. Wahba (1992). "Multivariate function and operator estimation, based on smoothing splines and reproducing kernels," In M. Casdagli and S. Eubank, editors, *Nonlinear Modeling and Forecasting, SFI Studies in the Sciences of Complexity, Proc. Vol XII*, pp. 95 – 112. Addison-Wesley.

G. Wahba and J. Wendelberger (1980). "Some new mathematical methods for variational objective analysis using splines and cross-validation," *Monthly Weather Review*, 108, 1122 – 1145.

G. Wahba and S. Wold (1975). "A completely automatic French curve," *Commun. Statist. Theory & Meth.*, 4, 1 – 17.

D. Wolpert (1992). “Stacked generalization,” *Neural Networks*, 5, 241 – 259.