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

TECHNICAL REPORT NO. 1015

February 28, 2000

Generalization and Regularization in Nonlinear Learning Systems [1]

by

Grace Wahba

[1]Prepared for the Handbook of Brain Theory and Neural Networks, Second Edition, Michael Arbib, Ed, within the space and reference limitations of the Handbook. This TR is an updated version of the entry of the same name in the First Edition, 1995, which was also printed as TR 921 1994. Supported by NIH Grant EY09946 and NSF Grant DMS9704758

Generalization and Regularization in Nonlinear Learning Systems

Grace Wahba
Department of Statistics
University of Wisconsin
1210 W. Dayton St.
Madison, WI 53706
wahba@stat.wisc.edu
February 24, 2000

1 Introduction

In this article we will describe generalization and regularization from the point of view of multivariate function estimation in a statistical context. Multivariate function estimation is not, in principle, distinguishable from supervised machine learning. However, until fairly recently supervised machine learning and multivariate function estimation had fairly distinct groups of practitioners, and small overlap in language, literature, and in the kinds of practical problems under study.

In any case, we are given a *training set*, consisting of pairs of input (feature) vectors and associated outputs $\{\boldsymbol{t}(i), y_i\}$, for n training or example subjects, $i = 1, ...n$. From this data, it is desired to construct a map which *generalizes well*, that is, given a new value of $\boldsymbol{t}$, the map will provide a reasonable prediction for the unobserved output associated with this $\boldsymbol{t}$.

Most applications fall into one of two broad categories, which might be called nonparametric regression and classification. In nonparametric regression, y may be (any) real number or a vector of r real numbers. The desired algorithm will produce an estimate $\hat{f}(\boldsymbol{t})$ of the expected value of a (new) y to be associated with a (new) attribute vector $\boldsymbol{t}$. In the (two-class) classification problem y_i will be an indicator whether or not the example (subject) came from class $\mathcal{A}$. In some classification applications, the desired algorithm will, given $\boldsymbol{t}$, return an indicator which predicts whether or not an example with attribute vector $\boldsymbol{t}$ comes from class $\mathcal{A}$ ('hard') classification. In other applications the desired algorithm will return $p(\boldsymbol{t})$, an estimate of the *probability* that the example with attribute vector $\boldsymbol{t}$ is in class $\mathcal{A}$. ('soft' classification). In some applications the feature vector $\boldsymbol{t}$ of dimension d contains zeroes and ones (for example as in a bitmap of handwriting), in others it may contain real numbers representing some physical quantities, ordered or unordered category indicators are also possible, as in medical demographic studies. *Regularization*, loosely speaking, means that while the desired map is constructed to approximately send the observed feature vectors to the observed outputs, constraints are applied to the construction of the map with the goal of reducing the generalization error. In some applications, these constraints embody *a priori* information concerning the true relationship between input and output; alternatively, various *ad hoc* constraints have sometimes been shown to work well in practice. Girosi, Jones and Poggio (1995) give a wide-ranging review.

2 Generalization and Regularization in Non-Parametric Regression

2.1 Single Input Spline Smoothing

We will use Figure 1 to illustrate the ideas of generalization and regularization in the simplest possible non-parametric regression setup, that is, $d = 1$, $r = 1$, with $\boldsymbol{t} = t$ any real number in some interval of the real line. The circles (which are identical in each of the three panels of Figure 1) represent $n = 100$ (synthetically

generated) input-output pairs $\{t(i), y_i\}$, generated according to the model

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

where $f_{TRUE}(t) = 4.26(e^{-t} - 4e^{-2t} + 3e^{-3t})$, and the ϵ_i came from a pseudorandom number generator for Normally distributed random variables with mean 0 and standard deviation $\sigma = 0.2$. Given this training data $\{t(i), y_i, i = 1, ..., n\}$, the learning problem is to create a map which, if given a new value of t, will predict the response $y(t)$. In this case, the data are noisy, so that even if the new t coincides with some predictor variable $t(i)$ in the training set, merely predicting y as the response y_i is not likely to be satisfactory. Also, this does not yet provide any ability to make predictions when t does not exactly match any predictor values in the training set. It is desired to generate a curve which will allow a reasonable prediction of the response for any t within a reasonable vicinity of the set of training predictors $\{t(i)\}$. The dashed line in each panel of Figure 1 is $f_{TRUE}(t)$; the three solid black lines in the three panels of Figure 1 are three solutions to the variational problem: Find f in the [Hilbert] space W_2 of functions with continuous first derivatives and square integrable second derivatives which minimizes

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(t(i))^2 + \lambda \int (f^{(2)}(u))^2 du, \tag{2}$$

for three different values of λ. The parameter λ is known as the regularization or smoothing parameter. As $\lambda \to \infty$, f_λ tends to the least squares straight line best fitting the data, and as $\lambda \to 0$ the solution tends to that curve in W_2 which minimizes the penalty functional $J(f) = \int (f^{(2)}(u))^2 du$ subject to interpolating the data (provided the $\{t(i)\}$ are distinct). This latter interpolating curve is known as a cubic interpolating spline, and minimizers of (2) are known as smoothing splines. See Wahba (1990) and references cited there for further information concerning these and other properties of splines noted below, and further references. In the top panel of Figure 1 λ has been chosen too small, and the wiggly solid line is attempting to fit the data too closely. It can be seen that using the wiggly curve in the top panel is not likely to give a good prediction of y, assuming that future predictor-response data is generated by the same mechanism as the training data. In the middle panel, λ has been chosen too large, the curve has been forced to flatten out, and again it can be seen that the heavy line will not give a good prediction of y. In the bottom panel, λ has been chosen by generalized cross validation (GCV). This is a method which behaves similarly to leaving-out-one in many cases but with computational and theoretical advantages. See Li(1986), Wahba(1990, Chapter 4), Girard(1998). It can be seen that the λ obtained this way does a good job of choosing the right amount of smoothing to best recover f_{TRUE} of Equation (1). The f_{TRUE} of Equation (1) would provide the best predictor of the response in an expected mean square error sense if future data were generated according to Equation (1). The curve in the bottom panel has a reasonable ability to *generalize*, that is, to predict the response given a new value t of the predictor variable, at least if t is not too far from the training predictor set $\{t(i)\}$.

For each positive λ, there exists a unique $\kappa = \kappa(\lambda)$ so that the minimizer f_λ of (2) is also the solution to the problem: Find f in W_2 to minimize

$$L(y, f) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(t(i))^2 \tag{3}$$

subject to the condition

$$J(f) = \int (f^{(2)}(u))^2 du \leq \kappa. \tag{4}$$

As λ becomes large, the associated $\kappa(\lambda)$ becomes small, and conversely. In general, the term *regularization* refers to solving some problem involving best fitting, subject to some constraint(s) on the solution. These constraints may be of various forms. When they involve a quadratic penalty involving derivatives, like $J(f)$,

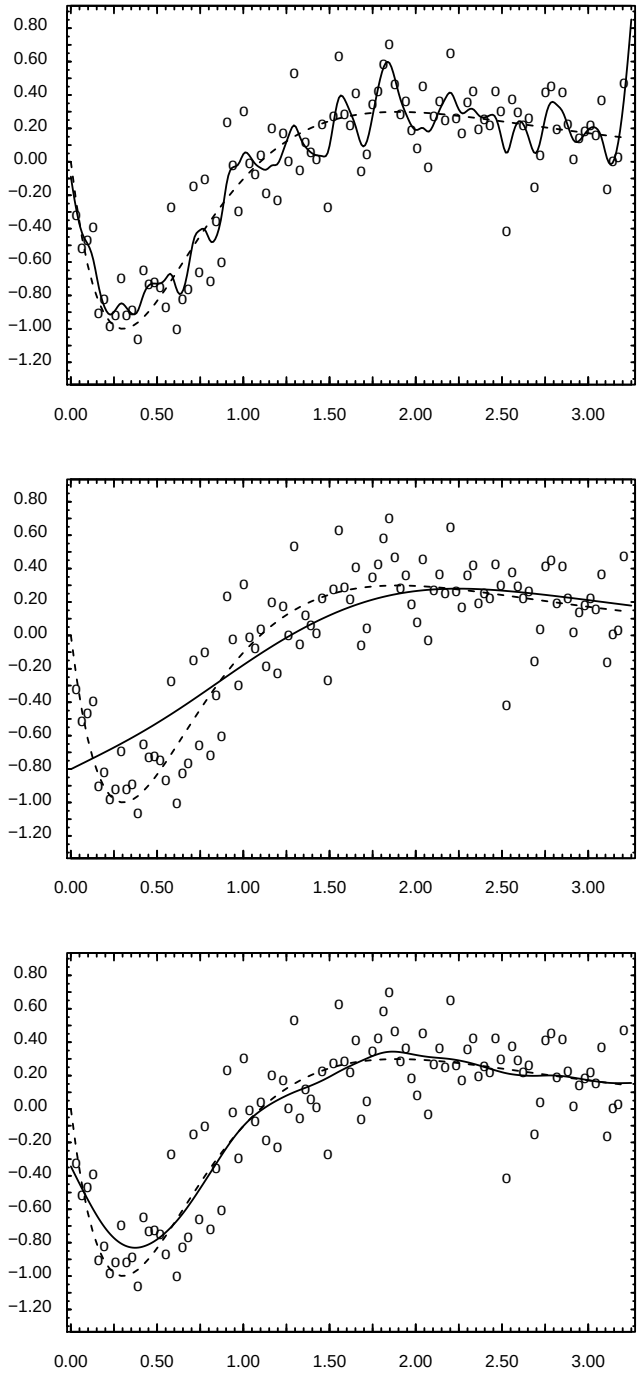

Figure 1: Training data (circles) have been generated by adding noise to $f_{TRUE}(t)$, shown by the dashed curve in each panel. All three panels have the same data. Top: Solid curve is fitted spline with λ too small. Middle: Solid curve is fitted spline with λ too large. Bottom: Solid curve is fitted spline with λ obtained by generalized cross validation.

the method is commonly referred to as Tikhonov regularization. The 'tighter' the constraints (i. e. the smaller κ, equivalently the larger λ) the further away the solution f_λ will generally be from the training data, that is, L will be larger. As the constraints get weaker and weaker then ultimately (if there are enough degrees of freedom in the method) the solution will interpolate the data. However, as is clear from Figure 1 a curve which runs through all the data points is *not* a good solution.

A fundamental problem in machine learning with noisy and or incomplete data, is to balance the 'tightness' of the constraints with the 'goodness of fit' to the data, in such a way as to minimize the 'generalization error', that is, the ability to predict the unobserved response for new values of t (or $\boldsymbol{t}$). This tradeoff is by now well known as the bias-variance tradeoff, or, equivalently, the goodness of fit - model complexity tradeoff. Methods abound in the statistical literature for univariate curve fitting, including Parzen kernel estimates, nearest neighbor estimates, orthogonal series estimates, least squares regression spline estimates, and, recently wavelet estimates. Each method has one or more regularization parameters, be they kernel window widths, numbers of nearest neighbors included, number of terms in the orthogonal series expansion or regression basis, or factors or thresholds for shrinking or truncating wavelet coefficients, that control this tradeoff. See Ramsay and Silverman (1997) and references cited there.

2.2 Multiple Input, Single Hidden Layer Feed-Forward Neural Net

A multiple input, single hidden layer feed-forward neural net (NN) predictor for the learning problem of Section 1 is typically of the form

$$f_{NN}(\boldsymbol{t}) = \sigma_0(b_o + \sum_{j=1}^{N} w_j \sigma_h(\boldsymbol{a}_j' \boldsymbol{t}(i) + b_j)) \tag{5}$$

where the $\boldsymbol{a}_j$ and $\boldsymbol{t}$ are d-vectors. The function σ_h is the so-called 'activation function' of the hidden layer and σ_0 is the activation function for the output. σ_h is generally a sigmoidal function, for example, $\sigma_h(\tau) = e^\tau/(1+e^\tau)$, while σ_0 may be linear, sigmoidal or a threshold unit. Here N is the number of hidden units, and the w_j, $\boldsymbol{a}_j$ and b_j are 'learned' from the training data by some appropriate iterative descent algorithm that tries to steer these values towards minimizing some distance measure, typically $L(y, f_{NN}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_{NN}(\boldsymbol{t}(i)))^2$. It is clear that if N is sufficiently large, and the descent algorithm is run long enough, it should be possible to drive the L as close as one likes to 0. (In practice it is possible to get stuck in local minima.) However, it is also clear intuitively from Figure 1 that driving L all the way to zero is not a desirable thing to do. Regularization in this problem may be done by controlling the size of N, by imposing penalties on the w_j, by stopping the descent algorithm early, that is, not driving down L as far as it can go, or by various combinations of these strategies. Each will influence how closely f_{NN} will fit the data, how 'wiggly' it will be, and how well it will be able to predict unobserved data that is generated by a similar mechanism as the observed data.

2.3 Multiple Input Radial Basis Function and Related Estimates

Radial basis functions are rapidly becoming a popular method for nonparametric regression. We first describe a general form of nonparametric regression which will specialize to radial basis functions and other methods of interest. Let $R(\boldsymbol{s}, \boldsymbol{t})$ be *any* symmetric, strictly positive definite function on $E^d \times E^d$. Here strictly positive definite means for any $K = 1, 2, ...$ the $K \times K$ matrix with j,kth entry $R(\boldsymbol{s}(j), \boldsymbol{s}(k))$ is strictly positive definite whenever the $\boldsymbol{s}(1), ..., \boldsymbol{s}(K)$ are distinct. (A symmetric $K \times K$ matrix M is said to be positive definite if for any K dimensional column vector x, $x'Mx$ is greater than or equal to 0, and is said to be strictly positive definite if $x'Mx$ is always strictly greater than 0.) Positive definiteness will play a key role in the discussion below because, (among other reasons) any positive definite matrix can be the covariance matrix of a random vector and any positive definite function $R(\boldsymbol{s}, \boldsymbol{t})$ can be the covariance function of some stochastic process, $X(\boldsymbol{t})$. That is, there exists $X(\cdot)$ such that $Cov\ X(\boldsymbol{s})X(\boldsymbol{t}) = R(\boldsymbol{s}, \boldsymbol{t})$. Given training data $\{\boldsymbol{t}(i), y_i\}$, it is always possible in

principle to obtain a (regularized) input-output map from this data by letting the model $f_{R,\lambda}$ be of the form

$$f_{R,\lambda}(\boldsymbol{t}) = \sum_{j=1}^{N} c_j R(\boldsymbol{t}, \boldsymbol{s}(j)), \tag{6}$$

where the $\boldsymbol{s}(j)$ are $N \leq n$ 'centers' which are placed at distinct values of the $\{\boldsymbol{t}(i)\}$ and $c = (c_1, ..., c_N)'$ is chosen to minimize $L(y, f) + \lambda J(f)$. Here

$$L(y, f_{R,\lambda}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_{R,\lambda}(\boldsymbol{t}(i))^2 \tag{7}$$

and the regularizing penalty $J(\cdot)$ is of the form

$$J(f_{R,\lambda}) = \sum_{j,k=1}^{N} c_j c_k J_{jk} \tag{8}$$

where J_{jk} are the entries of a non-negative definite quadratic form. The (strict) positive definiteness of R guarantees that

$$L(y, f_{R,\lambda}) + \lambda J(f_{R,\lambda}) \tag{9}$$

always has a unique minimizer in c, for any non-negative λ. This follows by substituting (6) into (9), and using the fact that the columns of the $n \times N$ matrix with i, j entry $R(\boldsymbol{t}(i), \boldsymbol{s}(j))$ are linearly independent since they are just N columns of the $n \times n$ positive definite matrix with i, j entry $R(\boldsymbol{t}(i), \boldsymbol{t}(j))$.

Radial basis function estimates are obtained for the special case where $R(\boldsymbol{s}, \boldsymbol{t})$ is of the special form

$$R(\boldsymbol{s}, \boldsymbol{t}) = r(\|W(\boldsymbol{s} - \boldsymbol{t})\|), \tag{10}$$

where W is some linear transformation on E^d and the norm is Euclidean distance. That is, $R(\boldsymbol{s}, \boldsymbol{t})$ depends only on some generalized distance in E^d between $\boldsymbol{s}$ and $\boldsymbol{t}$. The regularization, that is, the effecting of the tradeoff between goodness of fit to the data and 'smoothness' of the solution, is performed by reducing N, and/or increasing λ. The choice of W will also affect the 'wiggliness' of $f_{R,\lambda}$ in the radial basis function case. Alternatively, a model can be obtained by choosing N small and minimizing $L(y, f)$. In that case N and W are the smoothing parameters.

In the special case $N = n, \boldsymbol{s}(i) = \boldsymbol{t}(i)$, the $f_{R,\lambda}$ can (for *any* positive definite R) be shown to be Bayes estimates, see Kimeldorf and Wahba (1970), Wahba (1990). Arguments can be given to show that if n is large and $N < n$ is not too small, then they are good approximations to Bayes estimates, see Wahba (1990, Chapter 7). In the special case $J_{i,j} = R(\boldsymbol{t}(i), \boldsymbol{t}(j))$, the Bayes model is easy to describe and we do it here; it is:

$$y_i = X(\boldsymbol{t}(i)) + \epsilon_i, \tag{11}$$

with $X(\boldsymbol{t})$ a zero mean Gaussian stochastic process with covariance $EX(\boldsymbol{s})X(\boldsymbol{t}) = bR(\boldsymbol{s}, \boldsymbol{t})$ and the ϵ_i independent zero mean Gaussian random variables with common variance σ^2, and independent of $X(\boldsymbol{t})$. In this case, the minimizer $f_{R,\lambda}$ of $L(y, f) + \lambda J(f)$, evaluated at $\boldsymbol{t}$, is the conditional expectation of $X(\boldsymbol{t})$, given $y_1, ..., y_n$ provided that λ is chosen as σ^2/nb. In general, pretending that one has a prior and computing the posterior mean or mode will have a regularizing effect. The discussion above extends to symmetric positive definite functions on *arbitrary* domains for $\boldsymbol{t}$ including those mentioned in Section 1.

Thin plate splines in d variables (of order m) consist of radial basis functions plus polynomials of total degree less than m in d variables. ($2m - d > 0$ is required for technical reasons.) Letting $\boldsymbol{t} = (t_1, ..., d_d)$, the thin plate splines are minimizers (in an appropriate function space) of

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{t}(i))^2 + \lambda \sum_{\alpha_1 + \cdots + \alpha_d = m} \frac{m!}{\alpha_1! \cdots \alpha_d!} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left(\frac{\partial^m f}{\partial t_1^{\alpha_1} \cdots \partial t_d^{\alpha_d}} \right)^2 dt_1 \cdots dt_d. \tag{12}$$

Setting $d = 1, m = 2$ gives the cubic spline case discussed earlier. Note that there is no penalty on polynomials of total degree less than m, the thin plate splines with a particular choice of λ are Bayes estimates with an improper prior (that is, infinite variance) on the polynomials of total degree less than m, see Wahba (1990) and references cited there.

Related variations on regularized estimates include additive smoothing splines, which are of the form

$$f(\boldsymbol{t}) = \mu + \sum_{\alpha=1}^{d} f_\alpha(t_\alpha) \tag{13}$$

where μ and the f_α are the solution to a variational problem of the form: Find μ and $f_1, .., f_d$ in a certain function space to minimize

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{t}(i))^2 + \sum_{\alpha=1}^{d}\lambda_\alpha J_\alpha(f_\alpha). \tag{14}$$

The J_α may be of the form of J in Equation (4). Here, there is a *regularization parameter* for each component. See Hastie and Tibshirani (1990), Wahba(1990). These additive models generalize to smoothing spline analysis of variance (SS-ANOVA) models. In the SS-ANOVA models interaction terms of the form $f_{\alpha\beta}(t_\alpha, t_\beta), f_{\alpha\beta\gamma}(t_\alpha, t_\beta, t_\gamma)$, etc., which satisfy side conditions making them uniquely determined, are added to the representation in Equation (13), and corresponding penalty terms with regularization parameters are added in Equation (14). The f_α, etc, may be generalized to themselves being radial basis functions. Behind these models are positive definite functions which are built up via tensor sums and products of positive definite functions, See Gu and Wahba (1993), Wahba (1990), Wahba, Wang, Gu, Klein and Klein (1995).

Regression spline ANOVA models be obtained by setting the $f_\alpha, f_{\alpha\beta}$ etc. as linear combinations of a (relatively small) number of basis functions (usually splines). In this case the number of the basis functions is probably the most influential regularization parameter. These and similar methods again all have either explicit or implicit regularization parameters which govern the balance between the complexity of the model and the fit to the data - the bias-variance tradeoff.

The usual criteria for the generalization error when the fit involves minimizing the observed residual sum of squares is the expected (comparative) residual sum of squares for new data, $EL(y_{new}, f_\lambda) - \sigma^2 \equiv L(f_{TRUE}, f_\lambda)$. Here the y_{new} are new observations. Leaving out one, leaving out 10%, leaving out a 1/3 representative sample ('tuning set') and GCV ('in sample tuning') are popular methods for choosing the tuning parameters to minimize this criteria. Codes in Splus (smooth.spline()), SAS (tpspline), netlib(/gcv), Funfits (sreg, tps), R(smooth.Pspline, gss) and elsewhere are available for implementing the univariate spline, thin plate spline and additive and interaction (ANOVA) splines with GCV to choose single or multiple smoothing parameters. Netlib, Funfits and R are freeware. The smooth.Pspline code in R at `http://www.r-project.org` was used to generate Figure 1.

3 Generalization and Regularization in Soft Classification

Soft classification is a natural goal in certain kinds of demographic medical studies - for example suppose a large training set is available from a demographic study, consisting of observations $\{\boldsymbol{t}(i), y_i\}$ where y_i is an indicator (1 or 0) of the presence or absence of some disease in subject i at the end of the study, and $\boldsymbol{t}(i)$ is a vector of values of risk factors for this subject at the beginning of the study. With this kind of data, it is frequently of interest to make a 'soft' classification, that is, to estimate the *probability* $p(\boldsymbol{t})$ that a new subject with predictor vector $\boldsymbol{t}$ will contract the disease. A doctor, given this model, may advise new patients which risk factor(s) are important for them to control to reduce the probability of their contracting the disease. A regularized (that is, 'smooth') estimate for $p(\boldsymbol{t})$ is desirable. Regularized estimates can be obtained as follows. First, define

$$f(\boldsymbol{t}) = log[p(\boldsymbol{t})/(1 - p(\boldsymbol{t}))]. \tag{15}$$

f is known in the statistics literature as the log odds ratio, or logit. Then $p(\boldsymbol{t})$ is a sigmoidal function of $f(\boldsymbol{t})$, that is $p(\boldsymbol{t}) = e^{f(\boldsymbol{t})}/(1+e^{f(\boldsymbol{t})})$. We will get a regularized estimate for f. $L(y,f)$ of Equation (3) will be replaced by an expression more suitable for $0-1$ data, by using the likelihood for this data. To describe the likelihood, note that if y is a random variable with $Prob\ [y=1] = p$ and $Prob\ [y=0] = (1-p)$, then the probability density (or likelihood) $P(y,p)$ for y when p is true, is just $P(y,p) = p^y(1-p)^{(1-y)}$, this merely says $P(1,p) = p$ and $P(0,p) = (1-p)$. Thus, the likelihood for $y_1,...,y_n$ (assuming that the y_i are independent), is

$$P(y_1,...,y_n; p(\boldsymbol{t}(1),...,p(\boldsymbol{t}(n)) = \Pi_{i=1}^n p(\boldsymbol{t}(i))^{y_i}(1-p(\boldsymbol{t}(i))^{(1-y_i)}. \tag{16}$$

Substituting f for p in (16), taking the negative logarithm, gives the negative log likelihood $L(y,f)$ in terms of f:

$$-logP(y_1,...,y_n; f(\boldsymbol{t}(1),...,f(\boldsymbol{t}(n)) \equiv nL(y,f) = \sum_{i=1}^n [\log(1+e^{f(\boldsymbol{t}_{(i)})}) - y_i f(\boldsymbol{t}(i))]. \tag{17}$$

It is natural for $L(y,f)$ to replace $L(y,f)$ of (3) (7), (14) when y_i is restricted to 0 or 1, since $L(y,f_{TRUE})$ is (a multiple of) the negative log likelihood for y generated by a model with Gaussian noise like (1). A neural net implementation of soft classification would consist of finding $f_{NN}(\boldsymbol{t}) = logit p_{NN}(\boldsymbol{t})$ of the form of Equation (5) to minimize $L(y,f)$ of (17). If N is large enough, then, in principle, f_{NN} may be driven so that $p_{NN}(\boldsymbol{t}(i))$ is close to 1 if y_i is 1, and is close to 0 if y_i is 0. Again, it is intuitively clear that this is not desirable. As before, a regularized, or smooth f_{NN} can be obtained by controlling N, penalizing the w_i, stopping the iterative fitting early, or some combination of these.

Penalized likelihood estimates of f are obtained by minimizing $L(y,f) + J_\lambda(f)$ where $J_\lambda(f)$ is a penalty functional corresponding to those in Equations (2), (9), (12) or (14) and its generalizations. A popular definition for the generalization error is the (unobservable) comparative Kullback- Leibler distance of the estimate to the true probability distribution, which can be shown to be given by $EL(y_{new}, f_\lambda)) = L(p_{TRUE}, f_\lambda)$. An estimate of the λ which minimizes this criteria can be obtained by withholding a representative subset $y_{[left-out]}$ of the training set and choosing λ to minimize $L(y_{[left-out]}, f_\lambda)$. Leaving-out-one estimates are also possible but generally not feasible in this case. Generalized approximate cross validation (GACV) is a feasible insample method of choosing λ; based on a leaving-out-one argument, it has been shown in simulation studies to provide a good estimate of the minimizer of $L(p_{TRUE}, f_\lambda)$, see Wahba, Lin, Gao, Xiang, Klein and Klein (1999).

4 Generalization and Regularization in Hard Classification

In the hard classification problem (here we will consider only two classes for simplicity), we are only interested in estimating whether an example with vector $\boldsymbol{t}$ is in class $\mathcal{A}$ not. This is the typical situation in, for example character recognition, voice recognition, and other situations where it is known that the $\boldsymbol{t}$'s from the two classes being examined are generally well separated. In that case (assuming, for simplicity that the examples from the two classes are represented in the training set equally as is the future population of interest, and, that costs of misclassification are the same for both classes), then the optimum classifier (to minimize the expected cost) would be $\mathcal{A}$ if $p(\boldsymbol{t})$ is greater than one-half, and not $\mathcal{A}$ otherwise. Equivalently, the same rule can be implemented by examining the sign of the logit $f(\boldsymbol{t})$. Here we are identifying $\mathcal{A}$ with the 1's, and optimum is with respect to minimizing the expected cost of future misclassification. Unfortunately, in general it is neither desirable nor feasible to estimate the logit f directly by the methods of Section 3, because in the well separated case f takes on values near $\pm\infty$, and, if d and/or the sample size is large solving the penalized likelihood problem of Section 3 is likely to be numerically unstable. Recently, support vector machines (SVM's) have been shown to provide an excellent method for classification in this situation. See Burges (1998).

The support vector machine (SVM) is implemented coding the y_i as ± 1 according as the ith example is in $\mathcal{A}$ or not. Given a positive definite function $R(\boldsymbol{s},\boldsymbol{t})$, we find a function f of the form $f(\boldsymbol{t}) = b + \sum_{i=1}^n c_i R(\boldsymbol{t}, \boldsymbol{t}(i))$

by finding b and $c = (c_i, \cdots, c_n)$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n}(1 - y_i f(\boldsymbol{t}(i))_+ + \lambda \sum_{i,j} c_i c_j R(\boldsymbol{t(i)}, \boldsymbol{t(j)}) \tag{18}$$

where $(\tau)_+ = \tau$ for $\tau > 0$ and 0 otherwise. Letting f_λ be the minimizer of (18), the classification algorithm is: for a new attribute vector $\boldsymbol{t}$, assign $\mathcal{A}$ if $f(\boldsymbol{t}) > 0$ and not $\mathcal{A}$ if $f(\boldsymbol{t}) < 0$. Lin (1999) has demonstrated the remarkable result that, under general circumstances with appropriately chosen λ, the SVM estimate f_λ tends almost everywhere to either 1 or -1 and is an estimate of $signf_{TRUE} \equiv sign(p_{TRUE} - \frac{1}{2})$, which is exactly what is needed to carry out the optimum classification algorithm. A popular choice for $R(\boldsymbol{s}, \boldsymbol{t})$ is $R(\boldsymbol{s}, \boldsymbol{t}) = \exp -\frac{1}{\sigma^2}\|\boldsymbol{s} - \boldsymbol{t}\|^2$ where $\|\cdot\|$ is the Euclidean norm. In this choice of $R(\cdot, \cdot)$ the result may be sensitive to both σ and λ. As before, the λ and σ may be chosen by leaving out a representative subset of the observations and choosing λ and σ to minimize some measure of the generalization error. Here the natural choice for generalization error would be the misclassification rate. A version of GACV for SVM's, again based on a leaving-out-one argument, may be used as an insample method for choosing λ and σ, see Wahba, Lin and Zhang (1999). The generalization error target for the GACV is $E\frac{1}{n}\sum_{i=1}^{n}(1 - y_{inew} f_\lambda(\boldsymbol{t}(i)))_+$. However, $\frac{1}{2}E\frac{1}{n}\sum_{i=1}^{n}(1 - y_{inew} sign[f_\lambda(\boldsymbol{t}(i))])_+$ is the expected misclassification rate, so that to the extent that f_λ resembles $signf_\lambda$, this criteria will be appropriate for the generalization error.

5 Choosing How Much to Regularize

At the time of this writing, it is a matter of lively debate and much research how to choose the various regularization parameters. Leaving out a large fraction of the training sample for this purpose and tuning the regularization parameter(s) to best predict the left-out data (according to whatever criteria of best prediction is adopted) is conceptually simple, defensible, and widely used (this is called out-of-sample tuning). Successively leaving-out-one, successively leaving-out-10% , and the in-sample methods GCV and GACV are all popular. See also Ye (1998) who discusses in-sample tuning methods related to GCV in the Gaussian case which allow comparisons across different regularized estimates. In the Normally distributed observational error case, if the standard deviation of the observational error (σ in Equation (1))is known then unbiased risk estimates become available. See Li (1986), Wahba (1990) and references cited there. When there is a Bayesian model behind the regularization procedure, then maximum likelihood estimates may be derived, see Wahba (1985), although in order for these and other Bayes estimates to do a good job of minimizing the generalization error in practice, it is usually necessary that the priors on which they are based are realistic.

6 Which method is best?

Feedforward neural nets, radial basis functions, and various forms of splines all provide regularized or regularizable methods for estimating 'smooth' functions of several variables, given a training set $\{\boldsymbol{t}(i), y_i\}$: Which approach is best? Unfortunately, there is not, nor is there likely to be, a single answer to that question. The answer most surely depends on the particular nature of the underlying but unknown 'truth' , the nature of any prior information that might be available about this 'truth', the nature of any noise in the data, the ability of the experimenter to choose the various smoothing or regularization parameters well, the size of the data set, the use to which the answer will be put, and the computational facilities available. From a mathematical point of view, the classes of functions well approximated by neural nets, radial basis functions, additive and interaction splines (ANOVA splines) are not the same, although all of these methods have the capability of approximating large classes of functions. Of course, if a large enough data set is available, models utilizing all of these approaches may be built, and tuned, and then compared on data that has been set aside for this

purpose. In-sample tuning methods for comparison across different regularized estimates in the hard and soft classification contexts are an area of active research.

REFERENCES

Burges, C. (1998), 'A tutorial on support vector machines for pattern recognition', *Data Mining and Knowledge Discovery* **2**, 121–167.

Girard, D. (1998), 'Asymptotic comparison of (partial) cross-validation, GCV and randomized GCV in nonparametric regression', *Ann. Statist.* **126**, 315–334.

Girosi, F., Jones, M. & Poggio, T. (1995), 'Regularization theory and neural networks architectures', *Neural Computation* **7**, 219–269.

Gu, C. & Wahba, G. (1993), 'Semiparametric analysis of variance with tensor product thin plate splines', *J. Royal Statistical Soc. Ser. B* **55**, 353–368.

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

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

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

Lin, Y. (1999), Support vector machines and the Bayes rule in classification, Technical Report 1014, Department of Statistics, University of Wisconsin, Madison WI.

Ramsay, J. & Silverman, B. (1997), *Functional Data Analysis*, Springer.

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

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

Wahba, G., Lin, X., Gao, F., Xiang, D., Klein, R. & Klein, B. (1999), The bias-variance tradeoff and the randomized GACV, *in* M. Kearns, S. Solla & D. Cohn, eds, 'Advances in Information Processing Systems 11', MIT Press, pp. 620–626. Full oral presentation at NIPS 11.

Wahba, G., Lin, Y. & Zhang, H. (1999), Generalized approximate cross validation for support vector machines, or, another way to look at margin-like quantities, Technical Report 1006, Department of Statistics, University of Wisconsin, Madison WI. to appear, Advances in Large Margin Classifiers, A. Smola, P. Bartlett, B. Scholkopf and D. Schurmans, eds, MIT Press.

Wahba, G., Wang, Y., Gu, C., Klein, R. & Klein, B. (1995), 'Smoothing spline ANOVA for exponential families, with application to the Wisconsin Epidemiological Study of Diabetic Retinopathy', *Ann. Statist.* **23**, 1865–1895. Neyman Lecture.

Ye, J. (1998), 'On measuring and correcting the effects of data mining and model selection', *J. Amer. Statist. Assoc.* **93**, 120–131.