Computational Statistics (1999) 14:277–292

Regularization in skewed binary classification

Sauchi Stephen Lee

Division of Statistics, University of Idaho, Moscow, ID 83844, U.S.A.

Summary

Skewed binary classification concerns the assignment of a *new* unknown object to one of two populations, 0 or 1, on the basis of a q-dimensional vector $\mathbf{x} = (x_1, ...x_q)$, where one of the populations, for example population 0, is the prevalent class. Assignment rules are developed from learning samples of known objects, that is, objects known to come from each of the two populations. Since population 1 is the rare class, overfitting and generalization problems arise easily for many classification models. We propose an effective solution by assigning more weights to class 1. The idea is to produce *noisy* replicates of the rare cases while keeping the dominant class 0 cases unchanged. The classification models considered are: nearest neighbor method, neural networks, classification trees, and quadratic discriminant. Noisy replication of the rare cases was applied to three real world and simulated data sets. Encouraging results were obtained for all the classification models considered.

Keywords: ROC curve

1 Introduction

Regularization is a technique of avoiding overfitting to the training data and improving generalization to the test data by penalizing the fit via some "smoothness" criteria. Raviv & Intrator (1995) and Sietsma & Dow (1991)

showed that adding some noise during neural network training helps generalization. Although it has been mentioned by Ripley (1996), the idea of regularization by adding noise is new in the Statistics community and no serious work has been done in this direction. The success of such regularization in neural networks suggests that it is a technique also worth investigating in Statistics, especially in the context of skewed binary classification.

Skewed binary classification concerns the assignment of a *new* unknown object to one of two populations, 0 or 1, on the basis of a q-dimensional explanatory vector $\mathbf{x} = (x_1, ...x_q)$, where one of the populations, population 0, is the prevalent class. Assignment rules are developed from learning a training data set $T = \{(\mathbf{x}_i, y_i), i = 1, \cdots, n\}$, where $y = 0$ if the object comes from class 0 and $y = 1$ if it comes from class 1.

Let class 0 be the numerous class and class 1 be the rare class. Overfitting and generalization problems arise easily in such skewed binary classification because of the sparseness of $(\mathbf{x}_i, 1)$ in the training data. A natural way to overcome the problem of sparseness is to increase the occurrence of the rare cases by noisy replication of $(\mathbf{x}_i, 1)$ in the training data set many times, say k times, to become $(\mathbf{x}_i + \Delta\mathbf{x}_{i1}, 1), \cdots, (\mathbf{x}_i + \Delta\mathbf{x}_{ik}, 1)$, where $\Delta\mathbf{x}_{ij}, j = 1, \cdots, k$ are small random perturbations at $\mathbf{x}_i$. The numerous cases $(\mathbf{x}_i, 0)$ remain unchanged.

In this paper we will study the effect of replicating and adding noise to the rare cases during training in skewed binary classification for several classification models. The models considered are nearest neighbor method, neural networks, classification trees, and quadratic discriminant. We will briefly review the classification methods in section 2 along with an assessment of their predictive performance. In section 3 we implement the idea of replicating and adding noise during training in an algorithm and perform computer simulation experiments on three data sets which are available from the Information and Computer Science repository of the University of California at Irvine (ftp ics.uci.edu /pub/machine-learning-databases). Promising and encouraging results are obtained and are summarized in section 4. The reason why noisy replication works is explained in section 5 along with some concluding remarks for further research.

2 Methods

There are many ways to develop the assignment rules. In the case of binary classification, some methods assign $y = 0$ or 1 for a given $\mathbf{x}$, while others could be viewed as methods to estimate the conditional probability $f(\mathbf{x}) = P(y = 1|\mathbf{x}) = 1 - P(y = 0|\mathbf{x})$, where $\mathbf{x}$ is any point in the q-dimensional space of all possible explanatory vectors. The classification models considered in this paper are nearest neighbor method, neural networks, classification trees, and quadratic discriminant. The first model classifies y as either 0 or 1, and the other three estimate the conditional probability $P(y = 1|\mathbf{x})$. We give

a brief outline for each model; a detailed description can be found in many books, for example, Ripley (1996).

2.1 Models

2.1.1 k nearest neighbor (k-nn)

For a given *future* $\mathbf{x}$, the k nearest (in terms of the scaled Euclidean distance) training observations are found, and y is classified as either 0 or 1 by majority vote, with ties broken at random, from these known k neighbors in the training set. We selected $k = 1$, a standard choice.

2.1.2 Neural networks (NN)

There are many kinds of neural networks (see Hertz *et al.* 1991 for an introduction) but we restrict ourselves to only supervised feedforward single hidden layer neural networks with logistic output activation function in this paper. The estimate of $f(\mathbf{x})$ is

$$\hat{f}(\mathbf{x}) = \phi(w_0 + \sum_h w_h \phi(w_{0h} + \sum_{j=1}^{q} w_{jh} x_j)),$$

where w_0, w_h, w_{0h}, w_{jh} are the connection weights and $\phi(\theta) = \frac{1}{1+\exp(-\theta)}$. This type of network has q units at the input layer, h hidden units at the middle hidden layer, and 1 output unit at the output layer. Such networks are very general and it has been shown by many authors that any continuous function $f(\mathbf{x})$ can be approximated by these networks for sufficiently large numbers of hidden units. Backpropagation is the most commonly used training algorithm to estimate the weights but it is known to converge very slowly. We chose to use the Splus library `nnet` provided by Brian Ripley and is available at *Statlib* (http://lib.stat.cmu.edu/). Fundamentally, the library `nnet` uses the backpropagation algorithm, but it treats the training as an optimization problem which utilizes quasi-Newton optimizer to speed up the calculation. Each explanatory variable in the training and test data was normalized by subtracting its mean and dividing by its standard deviation. The training of the nets was stopped at the 500-th epoch. We chose a simple neural network with one single hidden layer and 2 hidden units (i.e., $h = 2$).

2.1.3 Classification and Regression Trees (CART)

A tree partitions the space of explanatory variables into locally constant regions, often hypercubes parallel to the variables axes. There are many different schemes for estimating trees. The idea is to recursively choose a variable or combination of variables and to split the variable's space on a carefully chosen value. The schemes differ in allowing multiway splits or restricting binary splits and in deciding how the best split is computed. Also,

they differ in when to stop growing the tree and how to prune it back for generalization. In this paper, we will use Breiman's CART (1984) which is best known and is commonly used in many disciplines. The conditional probability $f(\mathbf{x})$ is estimated to be the proportion of $y = 1$ observations among those in the terminal node containing the prediction point $\mathbf{x}$.

2.1.4 Quadratic Discriminant (QD)

This method estimates $f(\mathbf{x})$ via the Bayes theorem

$$f(\mathbf{x}) = P(y = 1|\mathbf{x}) = \frac{P(\mathbf{x}|y = 1)P(y = 1)}{P(\mathbf{x}|y = 0)P(y = 0) + P(\mathbf{x}|y = 1)P(y = 1)},$$

where $P(\mathbf{x}|y = i)$ is the probability density function of $\mathbf{x}$ for the population of class i, and $P(y = i)$ is the prior unconditional probability of class i, $i = 0, 1$. In Splus, these priors are not assumed to be uniform, instead, the class proportions for the training set are used to estimate the priors. The probability density function $P(\mathbf{x}|y = i)$ for class i, $i = 0$ and 1, is assumed to be a q-variate normal with mean μ_i and variance covariance matrix $\mathbf{\Sigma}_i$. That is to say,

$$P(\mathbf{x}|y = i) = \frac{1}{(2\pi)^{q/2}|\mathbf{\Sigma}_i|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \mu_i)'\mathbf{\Sigma}_i^{-1}(\mathbf{x} - \mu_i)\},\ i = 0, 1.$$

The parameters are estimated from the training data T and the estimated normal densities are substituted into the Bayes theorem to estimate $f(\mathbf{x})$.

2.2 Assessment via ROC curve

Since classification is a prediction problem, the performance of these assignment rules are measured in terms of the error rate for *future* $\mathbf{x}$ values. Let Ω be the entire space of explanatory variables; that is, the collection of all possible *future* $\mathbf{x}$ observations. Let R_0 be the decision region of *future* $\mathbf{x}$ values for which we classify objects as class 0, and $R_1 = \Omega - R_0$ be the remaining region for which we classify objects as class 1. Let $C_{i,j}$ be the misclassification cost of assigning an object as class j when, in fact, it is from class i. Let p_i be the prior probability of class i, and $g_i(\mathbf{x})$ be the probability density function of the q-dimensional explanatory vector $\mathbf{x}$ for the class i. If the misclassification costs could be specified, then the error rate could be found by

$$C_{0,1} * P(\text{misclassify a class 0 object}) + C_{1,0} * P(\text{misclassify a class 1 object})$$

$$= C_{0,1} * p_0 \int_{R_1} g_0(\mathbf{x})d\mathbf{x} + C_{1,0} * p_1 \int_{R_0} g_1(\mathbf{x})d\mathbf{x}.$$

This error rate depends on the specification of the misclassification costs and the prior probabilities which are subjective. To avoid the arbitrariness of

specifying these quantities, we will adopt the area under a Receiver Operative Characteristic (ROC) curve (Hanley and McNeil 1982) as the measure of predictive performance.

Let us call class 0 cases as negatives and class 1 cases as positives. An ROC curve is a plot of the true positive rate versus the false positive rate of a classification rule as the cut-off probability varies from 0 to 1. If the classification model estimates $f(\mathbf{x})$, then a case is classified as positive if the model outputs a $\hat{f}(\mathbf{x})$ value larger than or equal to the cut-off; otherwise, the case is classified as negative. The true positive rate is defined as the number of positives correctly classified divided by the total number of positives; the false positive rate is defined as the number of negatives incorrectly classified divided by the total number of negatives. An ideal model would have an area equal to 1.0 since the true positive rate is 1 and the false positive rate is 0 regardless of the cut-off value. One model is better than another with respect to this criterion if it has a larger ROC area.

3 Simulations

We applied the four classification models, namely, 1-nearest neighbor, neural net with 2 hidden units, CART, and quadratic discriminant, to the following three data sets which are publicly available from the UCI repository. The first two are real world data sets and the third is a simulated data.

3.1 Data sets

The following is a brief description of the three chosen data sets:

3.1.1 Diabetes

This is a data set gathered among the Pima Indians by the National Institute of Diabetes and Digestive and Kidney Diseases. The data set consists of 768 cases and 8 input variables which are medical information and physical measurements on each patient. The response variable y is one of two classes: tested positive for diabetes (268 cases) or negative (500 cases). Mutually disjoint training and validation data sets of the same size were randomly drawn from these 768 cases. To make the classification more skewed, we randomly selected 250 negatives and 15 positives in the training data set. The validation data set consists of the same number of positive and negative cases as the training data.

3.1.2 Hypothyroid

This is a data set with many qualitative and quantitative input variables and a lot of missing values. Since it does not make sense to add noise to qualitative variables, we just consider the five quantitative variables denoted

by TSH, T3, TT4, T4U, and FTI from the UCI repository. We cleaned up the data set by removing all missing values. After such data preprocessing, there are 2000 cases left consisting of 1878 class 0 (negative) cases and 122 class 1 (positive) cases. Mutually disjoint training and validation data sets of the same size were randomly drawn from these 2000 cases. To make the classification even more skewed, we randomly selected 900 negatives and 30 positives in the training data set. The validation data set consists of the same number of positive and negative cases as the training data.

3.1.3 Waveform

This is a simulated data set consisting of 5000 cases and 3 types of waves 0, 1, and 2, each having probability 1/3. It is described in Breiman *et al.* (1984) and a C subroutine for generating the data is in the UCI repository. To make it into a binary classification problem, we grouped the 3343 waves 0 and 1 to form the class 0 and the 1657 wave 2 to form the class 1. Mutually disjoint training and validation data sets of the same size were randomly drawn from these 5000 cases. To skew the classification, we randomly selected 1000 class 0 cases and 10 class 1 cases in the training data set. The validation data set consists of the same number of class 0 and 1 cases as the training data.

3.2 Simulation Algorithm

Simulation experiments were performed based on the following algorithm to add noise to the data sets:

- Initialization. Let $j = 1$.
  - **Step 1:** Training data set T_j and validation data set V_j are independently drawn without replacement from the data sets.
  - **Step 2:** Noisy replicates of rare cases $(\mathbf{x}_i, 1)$s in T_j are produced as follows:
    * Let $T_j = \{(\mathbf{x}_i, y_i), i = 1, \cdots, n\}$ be a training data set of size $n = n_0 + n_1$, where $\mathbf{x}$ is a q-dimensional vector of explanatory variables and y is a binary 0 and 1 (rare) response, with n_0 class 0 cases and n_1 class 1 cases, and $n_0 \gg n_1$.
    * Let *repl* and σ_{noise} be two fixed constants. When $y_i = 1$, replicate $(\mathbf{x}_i, y_i)$ *repl* times to become the set $\{(\mathbf{x}_i + \epsilon_{ik}, y_i)$: $\epsilon_{ik} \sim \mathbf{N}_q(\mathbf{0}, \sigma^2_{noise}\mathbf{\Sigma}_q)$, $k = 1, \cdots, repl\}$, where $\mathbf{\Sigma}_q$ is the $q \times q$ diagonal matrix $diag\{s_1^2, \cdots, s_q^2\}$, with s_l^2 the sample variance of the l-th explanatory variable x_l over the training data set. Note that the set becomes *repl* exact-copies of $(\mathbf{x}_i, y_i)$ when $\sigma_{noise} = 0$.
    * When $y_i = 0$, $(\mathbf{x}_i, y_i)$ remains unchanged.

  - The resulting training data set is denoted by T_j^*, with sample size increased from $(n_0 + n_1)$ to $(n_0 + repl * n_1)$.
- **Step 3:** Classification models are fitted to T_j and T_j^*, and the estimated models are denoted by $Model_{T_j}$ and $Model.noisy_{T_j^*}$, respectively.
- **Step 4:** Using the validation data set V_j, the predictive accuracy of the two models are measured and compared in terms of the areas under their ROC curves. Let A_j and A_j^* denote the respective ROC areas for $Model_{T_j}(V_j)$ and $Model.noisy_{T_j^*}(V_j)$. When the quantity $\Delta A_j = A_j^* - A_j$ is positive, the regularized model $Model.noisy$ predicts better than the unregularized model. When the quantity ΔA_j is negative, adding noisy replicates does not help in prediction.

- Let $nsim$ represents the number of times we repeat the experiment. If $j < nsim$, then $j = j+1$, and go to Step 1. Otherwise, stop and report $\{(A_j, A_j^*), j = 1, \cdots, nsim\}$.

There are three metaparameters in the above algorithm: $repl$, the number of noisy replicates generated for a given rare case; σ_{noise}, the standard deviation of the noise; and $nsim$, the number of pairs of training data set and validation data set generated in the simulation. We chose $repl = 2$ which corresponds to the smallest non-trivial number of replication and $nsim = 500$. For the more critical metaparameter σ_{noise}, we did an empirical study for various σ_{noise} values started from zero and increased in steps of .5 until the performance deteriorated. Let $\mu_{\Delta A, \sigma_{noise}}$ denote the true unknown mean change in ROC area when noisy replicates of magnitude σ_{noise} are added during training. We computed a 95% confidence interval for $\mu_{\Delta A, \sigma_{noise}}$ via the formula

$$\overline{\Delta A} \pm 1.96 * \frac{\hat{\sigma}_{\Delta A}}{\sqrt{nsim}},$$

where

$$\overline{\Delta A} = \frac{\sum_{j=1}^{nsim} \Delta A_j}{nsim} = \frac{\sum_{j=1}^{nsim} A_j^* - A_j}{nsim}$$

and $\hat{\sigma}_{\Delta A}$ is the standard deviation of $\{\Delta A_j, j = 1, \cdots, nsim\}$.

If there is no difference in adding noisy replicates during training, then $\mu_{\Delta A, \sigma_{noise}} = 0$ and the constructed interval should contain the point 0. If the addition of noisy replicates during training improves the generalization, then $\mu_{\Delta A, \sigma_{noise}} > 0$ and the constructed confidence interval should not contain 0 and the entire interval should be positive, and vice versa. The confidence intervals were plotted in Figures 1, 2, and 3, which correspond to the three data sets Diabetes, Hypothyroid, and Waveform, respectively. The four classification models were fitted to all three data sets.

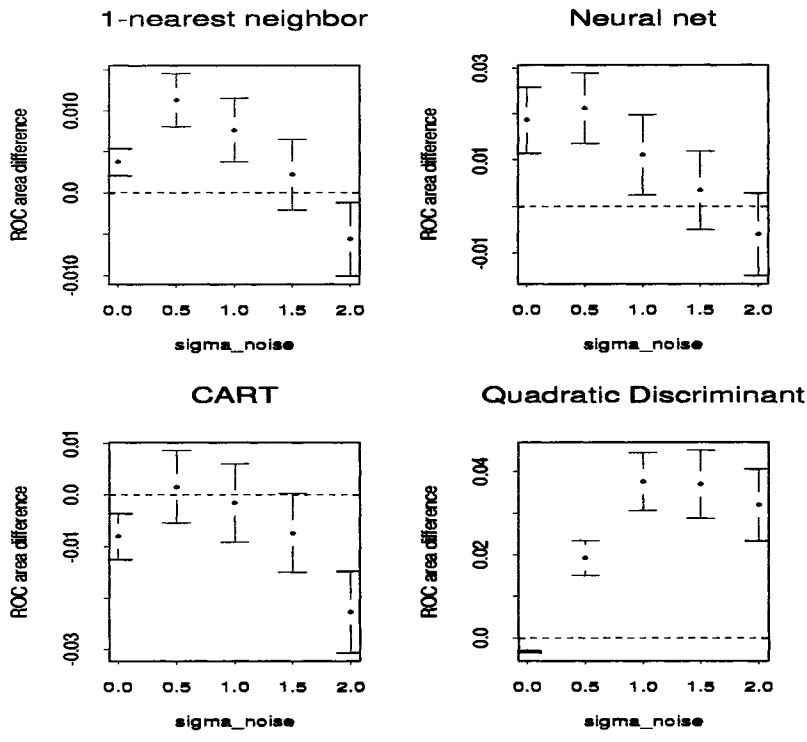

Figure 1: 95% Confidence Intervals of the ROC area difference for the Diabetes Data as σ_{noise} is increased from 0 in steps of 0.5

It is clear from the figures that some confidence intervals lie above the x-axis, indicating that the addition of noisy replicates during training could produce better predictions for some σ_{noise} values.

4 Results

The optimal σ_{noise} was selected when the performance, with respect to the ROC area, of the *noisy* model just started to go downhill. Let $\mu_{ROC,original}$ and $\mu_{ROC,opt-noise}$ denote the true mean ROC area for the original (no noisy replicates) model and the optimal-noise model, respectively. Table 1 summarizes the results and the meaning of the columns' label are as follows: **Data set**, the name of the data set; **Model**, the classification model; **Opt.** σ_{noise}, the optimal noise standard deviation; **Mean ROC area: Orig.** *vs.* **Opt.**, the average of the 500 ROC areas for the original model versus the

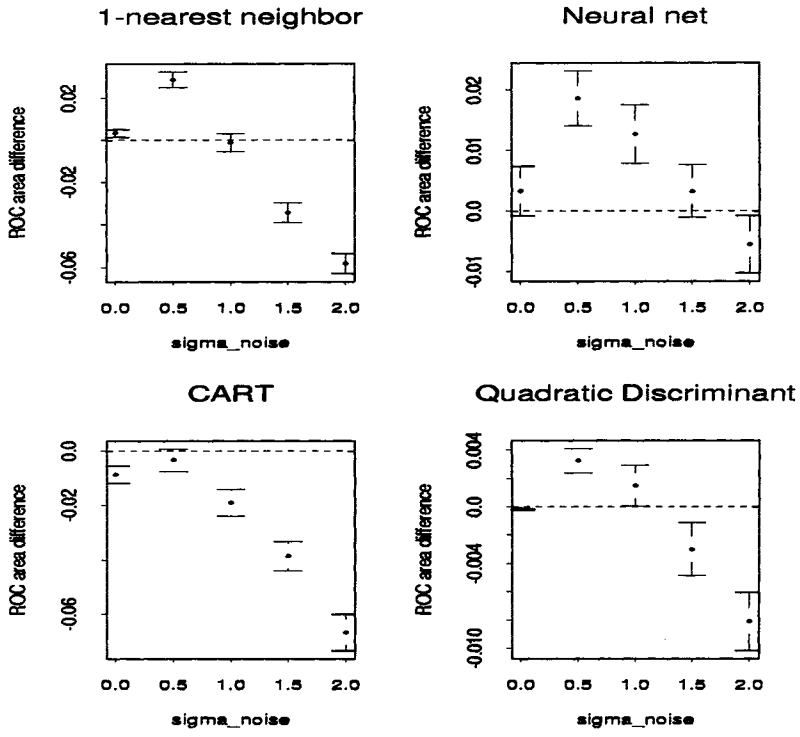

Figure 2: 95% Confidence Intervals of the ROC area difference for the Hypothyroid Data as σ_{noise} is increased from 0 in steps of 0.5

optimal-noise model (the symbol '<' indicate the quantity on the left is significantly less than the quantity on the right, the symbol '≈' indicate the two quantities are not significantly different); **p-value**, the two-tailed p-value for testing $H_0 : \mu_{ROC,original} = \mu_{ROC,opt-noise}$ against $H_a : \mu_{ROC,original} \neq \mu_{ROC,opt-noise}$ using a paired-sample t test; and **% change**, the percentage increase or decrease in mean ROC area of the optimal-noise model relative to the mean ROC area of the original model (i.e., % change = {mean ROC area of the optimal-noise model - mean ROC area of the original model}/mean ROC area of the original model). Boxplots of the 500 ROC areas obtained from the original and optimal-noise models fitted to the three data sets are presented in Figures 4, 5, and 6.

From Table 1 and Figures 4 to 6, we observe the following:

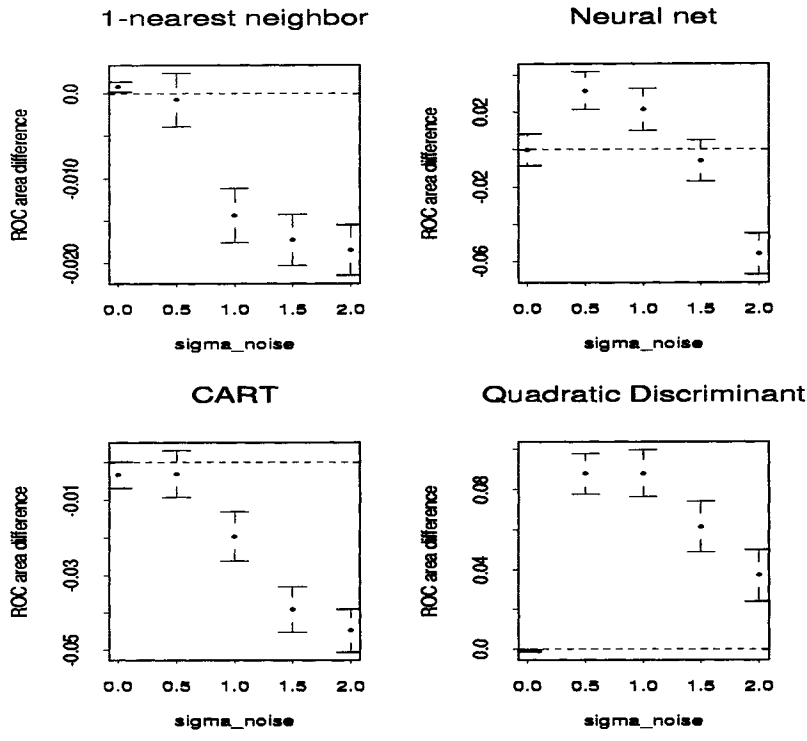

Figure 3: 95% Confidence Intervals of the ROC area difference for the Waveform Data as σ_{noise} is increased from 0 in steps of 0.5

- 1-nearest neighbor: All three data sets had significant increase in the mean ROC area. The largest percentage increase in mean ROC area was 4.1%.
- Neural Nets: All three data sets had significant increase in the mean ROC area. The largest percentage increase in mean ROC area was 4.8%.
- CART: All three data sets had no significant change in the mean ROC area. This could be improved as explained in section 5.
- Quadratic discriminant: All three data sets had significant increase in the mean ROC area. The largest percentage increase in mean ROC area was 15%.

Table 1: Summary of the 500 ROC areas for the original and optimal-noise models.

Data set	Model	Opt. σ_{noise}	Mean ROC area: Orig. *vs.* Opt.	p-value	% change
Diabetes	1-nn	0.5	.544 < .556	< .0001	2.2%
	NN	0.5	.670 < .692	< .0001	3.3%
	CART	0.5	.621 ≈ .623	.6600	0.3%
	QD	1.0	.629 < .667	< .0001	6.0%
Hypo-thyroid	1-nn	0.5	.712 < .741	< .0001	4.1%
	NN	0.5	.870 < .889	< .0001	2.2%
	CART	0.5	.873 ≈ .870	.1100	-0.3%
	QD	0.5	.898 < .902	< .0001	0.5%
Wave-form	1-nn	0.0	.518 < .519	.0074	0.2%
	NN	0.5	.645 < .676	< .0001	4.8%
	CART	0.5	.552 ≈ .549	.3200	-0.5%
	QD	1.0	.585 < .673	< .0001	15.0%

5 Discussion

The improvement in prediction by adding noisy replicates during training could be explained by the trade off between bias and variance of the estimates. It is clear that adding noise will increase the bias of the estimator. If the decrease in variance is even larger, then this will result in a better prediction.

This is clearly the case for the nearest neighbor method and the improvement is encouraging for the three data sets.

It is known that neural networks and CART have a tendency to easily overfit the training data. Regularization by adding noisy replicates to neural network inputs does indeed help to improve prediction as expected. The result for CART does not turn out to be as good as we expected. However, it does not surprise us because the structure of CART is very sensitive to noise perturbation. In other words, a slight change in $\mathbf{x}$ will lead to a drastically different tree. We have started a new research direction in which we are averaging multiple copies of noise-generated trees. We are anticipating that such regularized CART will show a more positive improvement because averaging different trees will significantly decrease the variance of the estimates (Breiman 1996).

The improvement to quadratic discriminant analysis obtained by adding noisy replicates is very encouraging. The increase in ROC area was very significant. Quadratic discriminant analysis is being commonly used in many disciplines, especially in the medical community where it is used to discriminate and diagnose various diseases. In many cases, the situation is a skewed binary classification.

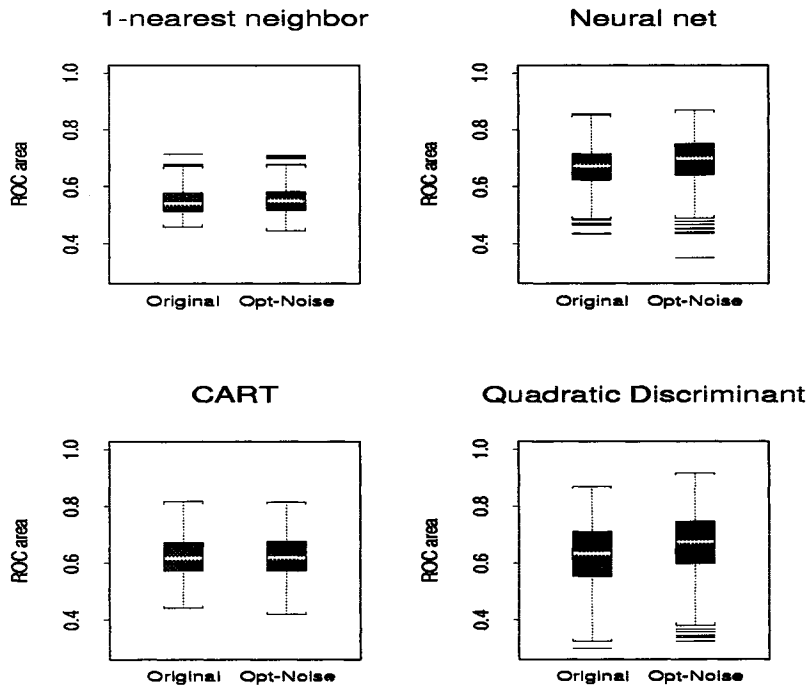

Figure 4: Boxplots for the 500 ROC areas obtained from the original and optimal-noise models for the Diabetes Data

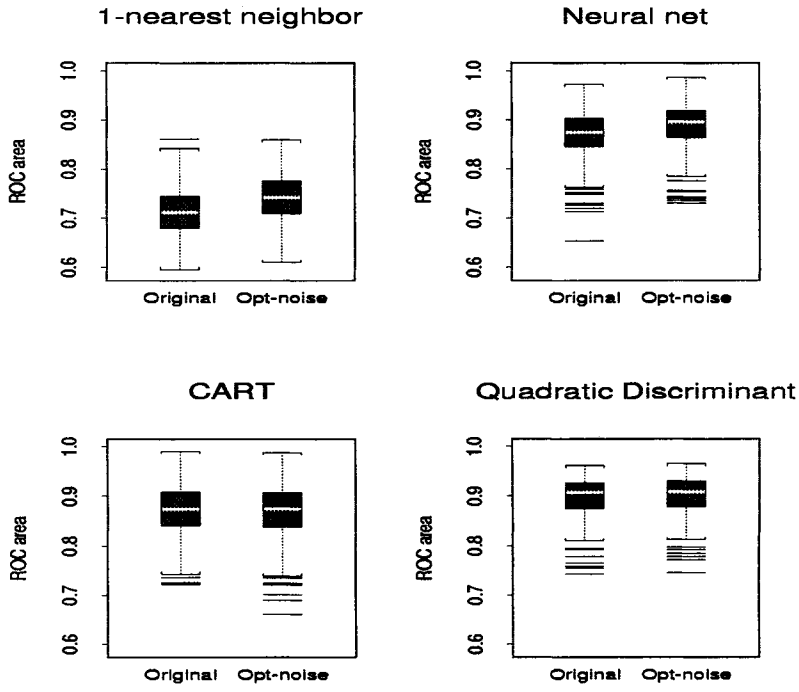

Figure 5: Boxplots for the 500 ROC areas obtained from the original and optimal-noise models for the Hypothyroid Data

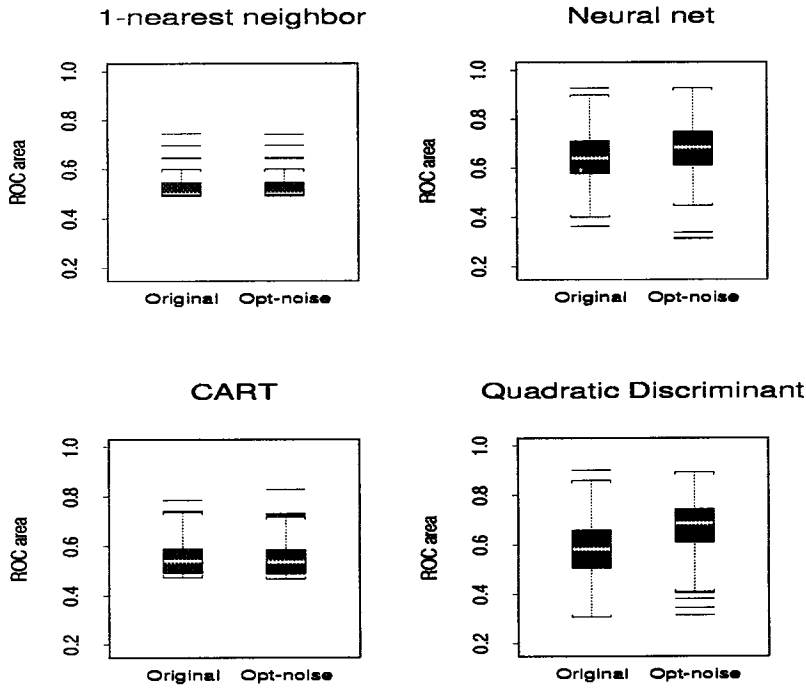

Figure 6: Boxplots for the 500 ROC areas obtained from the original and optimal-noise models for the Waveform Data

Many forms of regularization exist for different models and they are model-specific. For example, neural networks could be regularized via weight decay (Venables and Ripley 1994), and CART could be regularized through pruning (Venables and Ripley 1994) according to some criteria, say, AIC, and many regularization techniques exist for quadratic discriminant (Mkhadri *et al.* 1997). Adding noisy replicates in skewed binary classification offers a simple, elegant, and unified treatment for regularization which is model-free.

To conclude, we found that adding noisy replicates to skewed binary classification is a promising and natural form of regularization. It is our hope that the success demonstrated for the models studied in this paper will provide a basis for further research. One potentially fruitful area worth investigation is the averaging of different versions of the noise-generated models $Model.noisy_{T^*}$ for a given training data set T. Furthermore, we assumed equal noise variance for the two classes and the variance covariance matrix of the noise $\mathbf{\Sigma}_q$ is diagonal; room for improvement exists if the variance equality assumption is relaxed and $\mathbf{\Sigma}_q$ is carefully chosen so that the correlations between the explanatory variables are taken into account. It might also be interesting to try some other nonparametric flexible models such as general additive model GAM and projection pursuit regression.

6 References

Bishop, C. (1995). Training with noise is equivalent to Tikohonov regularization. *Neural Computation*, **7**, pp.108-116.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, **26**, No. 2, pp.123-140.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees.* Wadsworth & Brooks, Monterey, California.

Hanley, J.A. and McNeil, B.J. (1982). The meaning and use of the area under a receiver operating characteristics (ROC) curve. *Radiology*, **143**, pp.29-36.

Hertz, J., Krogh, A., and Palmer, R.G. (1991). *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, CA.

Mkhadri, A., Celeux G., and Nasroallah A. (1997). Regularization in discriminant analysis: An overview. *Computational Statistics and Data Analysis*, **23**, pp.403-423.

Quinlan, J. R. (1993). *C4.5: Program for Machine Learning.* Morgan Kaufmann, San Mateo.

Raviv, Y. and Intrator, N. (1995). Bootstrapping with noise: An effective regularization technique. Technical Report, Tel-Aviv University, Israel.

Ripley, B.D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press.

Sietsma, J. and Dow, R.J.F. (1991). Creating artificial networks that generalize. *Neural Networks*, **4**, pp.67-79.

Venables, W.N. and Ripley, B.D. (1994). ***Modern Applied Statistics with S-plus***. Springer-Verlag, New York.