ELSEVIER Computational Statistics & Data Analysis 34 (2000) 165–191

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Noisy replication in skewed binary classification

Sauchi Stephen Lee

Division of Statistics, University of Idaho, Moscow, ID 83844, USA

Received 1 October 1998; received in revised form 1 September 1999

Abstract

Skewed binary classification problems arise in estimating the "success" probabilities of *new* observations due to sparse "successes" and numerous "failures" in a given training data set. Previously Lee (1999) showed that adding small normal noise to replicate the "successes" in the training set could slightly improve estimates in several common classification models, namely, nearest neighbor, neural networks, classification trees, and quadratic discriminant. Now, we form much improved estimates for these same models: generating multiple versions of noise-added training sets from a given data set, we obtain an average of multiple model estimates. This model average is significantly improved both in terms of ROC area and Kullback–Leibler distance. In effect, the technique serves as an effective and model-free regularization for the classification models considered. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: ROC curve; Kullback–Leibler distance

1. Introduction and summary

Skewed binary classification concerns the assignment or prediction of a *new* unknown object to one of two populations, 0 or 1, on the basis of a q-dimensional explanatory vector $\boldsymbol{x} = (x_1, \dots, x_q)$ measured on the object, where one of the populations, population 0, is the prevalent class. Let the variable y represent the class label with $y = 0$ if the object comes from population 0 and $y = 1$ if it comes from population 1. Assignment rules are developed from modeling a training data set $T = \{(\boldsymbol{x}_i, y_i),\ i = 1, \dots, n_0 + n_1\} = \{(\boldsymbol{x}_i, 0),\ i = 1, \dots, n_0\} \cup \{(\boldsymbol{x}_i, 1),\ i = 1, \dots, n_1\}$, where $n_0 \gg n_1$, that is, the number of class 0 cases is much larger than the number of class 1 cases.

PII: S 0167-9473(99)00095-X

Overfitting and generalization problems easily arise in estimating the probability $f(\boldsymbol{x}_{\text{new}})=P(y=1|\boldsymbol{x}_{\text{new}})$ of future observation $\boldsymbol{x}_{\text{new}}$ because of the sparseness of $(\boldsymbol{x},1)$ in the training data. A natural way to overcome the problem of sparseness of class 1 cases is to increase their occurrence by replicating $(\boldsymbol{x},1)$ in the training data set many times, say m times, where m is a constant number. The replicates are chosen to be of the form $(\boldsymbol{x}+\varepsilon_j,1)$, $j=1,\dots,m$, where ε_j is a small normal noise. In short, we will call these replicates *noisy replicates*. As the variance of the noise tends to zero, the noisy replicates $(\boldsymbol{x}+\varepsilon_j,1)$, $j=1,\dots,m$, would become exact copies of $(\boldsymbol{x},1)$. The numerous cases $(\boldsymbol{x},0)$ remain unchanged.

Let T^* denote the resulting noisy training data set when the original rare cases in T are replaced by *noisy replicates*. In other words, $T^*=\{(\boldsymbol{x}_i,0),\ i=1,\dots,n_0\}\cup\{(\boldsymbol{x}_i+\varepsilon_{ij},1),\ i=1,\dots,n_1;\ j=1,\dots,m\}$. Note that the sample size of the noisy training data set has increased from n_0+n_1 to n_0+m*n_1, and the skew between class 0 and class 1 cases has decreased. Let $\hat{f}_T(\boldsymbol{x})$ be an estimate of $f(\boldsymbol{x})$, via a certain classification model, based on a training data set T. Let $\hat{f}_{T^*}(\boldsymbol{x})$ be an estimate of $f(\boldsymbol{x})$, via the same classification model, based on a noisy training data set T^* generated from T. It has been shown (Lee, 1999) that $\hat{f}_{T^*}(\boldsymbol{x})$ can be slightly better than $\hat{f}_T(\boldsymbol{x})$ in four commonly used classification models: nearest neighbor, neural networks, classification trees, and quadratic discriminant. In the present paper we will extend the methodology by averaging multiple versions of $\hat{f}_{T^*}(\boldsymbol{x})$. For a given training data set T, we will independently generate several noisy training data sets T_k^*, $k=1,2,\dots$, and will average the $\hat{f}_{T_k^*}(\boldsymbol{x})$, $k=1,2,\dots$, to obtain a final estimate of $f(\boldsymbol{x})$, which is denoted by $\hat{f}_{\bar{T}^*}(\boldsymbol{x})$. This estimate, $\hat{f}_{\bar{T}^*}(\boldsymbol{x})$, is found to be much better than both $\hat{f}_T(\boldsymbol{x})$ and $\hat{f}_{T^*}(\boldsymbol{x})$ in terms of ROC area and Kullback–Leibler distance (see below).

The roles of the noisy training data set T^* are two fold. First, it increases the sample size of the rare class and thus decreases the skew between the two classes. Second, it pushes different estimates $\hat{f}_{T^*}(\boldsymbol{x})$ of $f(\boldsymbol{x})$ to different local optima depending on T^*, and by that produces a set of biased and relatively independent estimates $\hat{f}_{T_1^*}(\boldsymbol{x})$, $\hat{f}_{T_2^*}(\boldsymbol{x}),\dots$. By averaging over these relatively independent estimators, the variance of $\hat{f}_{\bar{T}^*}(\boldsymbol{x})$ will be decreased. We found that the trade off between the bias and the variance results in better predictions for the classification models considered in this paper. In other words, we discovered that the addition of noisy replicates serves as an effective regularization technique.

Various regularization methods in classification has currently appeared in the literature. Breiman (1996,1998) used a method called "bagging/arcing" to bootstrap B times a given training data set of n cases with equal/unequal probabilities on each case to generate B classifiers, and combined them by simple voting. Freund and Schapire (1996) proposed a boosting algorithm to generate and combine classifiers so as to drive the training data set classification error to zero as quickly as possible. Raviv and Intrator (1996) showed that adding some noise during neural networks training results in a better classification for *new* objects for the deterministic "Two-Spiral" pattern recognition problem. Although it has been mentioned by Ripley (1996), the idea of regularization by adding noise is still quite new in the Statistics community and little work has been done in this direction. The initial success of such regularization by adding noise (Lee, 1999) encouraged us to further investigate the technique in the context of skewed binary classification. To our excitement, we found that adding suitable amount of noise and averaging multiple $\hat{f}_{T^*}(\boldsymbol{x})$s will significantly improve the classification rate for the classification models considered.

This paper is organized as follows. We will briefly review several common classification methods in Section 2 and introduce two measures of predictive performance. In Section 3 the idea of noisy replication during training is described and implemented in an algorithm. This algorithm also allowed us to choose the optimal amount of noise to be added. Computer simulation experiments were conducted on two simulated data sets and two real-data sets. The two simulated data sets included a standard stochastic task of classifying two bivariate normals, and a financial problem generated from a set of deterministic rules. The two real-data sets are medical diagnosis problems and are available from the Information and Computer Science repository of the University of California at Irvine. Despite the different nature of the data sets and the different structure of the classification models, significant positive results are obtained for all the data sets and classification models considered. The results are summarized in Section 4. Discussion and concluding remark for further research are considered in Sections 5 and 6.

2. Classification models

Given a training data set $T=\{(\boldsymbol{x}_i, y_i),\ i=1,\dots,n=n_0+n_1\}=\{(\boldsymbol{x}_i,0),\ i=1,\dots,n_0\}\cup\{(\boldsymbol{x}_i,1),\ i=1,\dots,n_1\}$, there are many ways to develop the assignment rules for future unknown object with explanatory vector $\boldsymbol{x}$. In the case of binary classification, they could be viewed as methods to estimate the conditional probability $f(\boldsymbol{x})=P(y=1|\boldsymbol{x})=1-P(y=0|\boldsymbol{x})$, where $\boldsymbol{x}$ is any point in the q-dimensional space of all possible explanatory vectors. We give a brief outline for each model: nearest-neighbor method, neural networks, classification trees, quadratic discriminant. Detailed descriptions of them can be found in many books, for example, Ripley (1996).

2.1. Models

2.1.1. k nearest neighbor

The standard k nearest-neighbor (k-nn) method estimates $f(\boldsymbol{x})=P(y=1|\boldsymbol{x})$ by

$$\hat{f}(\boldsymbol{x})=\frac{1}{k}\sum_{i=1}^{n}\mathbf{1}(||\boldsymbol{x}-\boldsymbol{x}_i||\le \mathcal{O}_k(\boldsymbol{x}))\mathbf{1}(y_i=1),$$

where the distance $||\boldsymbol{x}-\boldsymbol{x}_i||$ is equal to $\sqrt{(\boldsymbol{x}-\boldsymbol{x}_i)'(\boldsymbol{x}-\boldsymbol{x}_i)}$, $\mathcal{O}_k(\boldsymbol{x})$ is the kth-order statistic of $\{||\boldsymbol{x}-\boldsymbol{x}_i||\}_1^n$, and $\mathbf{1}(\cdot)$ is an indicator function such that $\mathbf{1}(\omega)=1$ if ω is true and 0 otherwise. The method estimates $f(\boldsymbol{x})$ as the proportion of class 1 (i.e., $y=1$) training observations among the k closest neighbors of $\boldsymbol{x}$. We chose the smallest possible value of k, namely, $k=1$, for the simulation experiments in Section 3.

2.1.2. Neural networks

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

$$\hat{f}(\boldsymbol{x}) = \phi\left(\hat{w}_0 + \sum_h \hat{w}_h \phi\left(\hat{w}_{0h} + \sum_{j=1}^{q} \hat{w}_{jh} x_j\right)\right),$$

where $\hat{w}_0, \hat{w}_h, \hat{w}_{0h}, \hat{w}_{jh}$ are the connection weights and $\phi(\theta) = 1/(1+\exp(-\theta))$. This type of networks 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 we denote them by the notation $q - h - 1$ NN. It has been shown by many authors that, for sufficiently large h, any continuous real-valued function $f(\boldsymbol{x})$ in the q-dimensional space can be approximated by these $q-h-1$ neural networks to any desirable degree of accuracy. Many numerical software packages exist to find the connection weights for a given training data set, we chose to use the Splus library `nnet` provided by Brian Ripley and is available at *Statlib* (http://lib.stat.cmu.edu/). The maximum number of training iteration to find the connection weights in `nnet` is set to be 100 epochs by default. We increased it to 400 epochs at the expense of more computing time to ensure numerical convergence. Since neural networks are very flexible nonparametric models, we will use the smallest possible non-trivial neural net, $q - 2 - 1$ NN, to minimize overfitting. The $q - 1 - 1$ neural net will collapse to logistic regression.

2.1.3. Classification trees

A tree partitions the q-dimensional space of explanatory variables into locally constant regions, often hypercubes parallel to the variables axes. There are many different schemes for estimating trees. The basic idea is to recursively choose a variable or combination of variables and to split the variable's space on a carefully chosen value. These 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. The conditional probability $f(\boldsymbol{x})$ is estimated to be the proportion of $y=1$ observations among those in the terminal node containing the prediction point $\boldsymbol{x}$. In this paper, we will use the Splus tree classifier which is based on the well-known Breiman's CART (1984). For a given training data set, we chose to fit two kinds of trees: a full-grown tree with no pruning, and a pruned tree obtained from the full-grown tree by snipping off the least important splits according to a cost-complexity factor (Venables and Ripley, 1994).

2.1.4. Quadratic discriminant

Quadratic discriminant (QD) method estimates $f(\boldsymbol{x})$ via the Bayes formula

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

where $P(\boldsymbol{x}|y=i)$ is the probability density function of $\boldsymbol{x}$ for the population of class i, and $P(y=i)$ is the prior unconditional probability of class i, $i=0,1$. The probability density function $P(\boldsymbol{x}|y=i)$ for class i, $i=0$ and 1, is assumed to be a q-variate normal with mean μ_i and variance covariance matrix $\boldsymbol{\Sigma}_i$. That is to say,

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

The parameters μ_i and $\boldsymbol{\Sigma}_i$ are estimated from the training data T and the estimated normal densities $P(\boldsymbol{x}|y=i)$ are substituted into the Bayes formula to estimate $f(\boldsymbol{x})$. The prior probability function is the porportion of $y=1$ in the original training data set T.

2.2. Prediction assessment

We would evaluate the classification performance using two different kinds of measures: discrimination and calibration. Many other measures, including misclassification rate, Brier score, sensitivity, specificity, and Gini (concordance) index, are closely related to these two types of fundamental measures (Hand, 1997, Chapter 6). The measure of discrimination (some call it separability, see Hand, 1997) refers to the capability of the model to distinguish correctly the two classes. Perfect discrimination means that the two classes could be separated into two non-overlapping sets of model predicted probabilities. The measure of calibration (some call it precision, see Hand, 1997) describes the closeness of the model's predicted probabilities to their target classes 0 and 1.

2.2.1. ROC area

A common measure of discrimination is the area under a receiver operating characteristic (ROC) curve (Hanley and McNeil, 1982). Let us call class 0 cases as negatives and class 1 cases as positives. A new case is classified as positive if a classification model outputs a $\hat{f}(\boldsymbol{x})$ value larger than or equal to a pre-chosen threshold value; otherwise, the case is classified as negative. An ROC curve is a plot of the true positive rate versus the false positive rate of a classification rule as the threshold value varies from 0 to 1. 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 ROC area equal to 1.0 (completely separable) since the true positive rate is 1 and the false positive rate is 0 regardless of the threshold value. By comparing ROC areas we can define a dominance relationship between classifiers. This dominance relationship is clear when the ROC curve from one model is always above the other and the two curves do not intersect. When they do intersect, one model is superior in some regions and another elsewhere. The total area under the curve becomes an average collective overall comparison between models. For example, in Fig. 1, it is said that classifier 1 is a *better* model than classifier 2 because classifier 1 yields a larger area under its ROC curve.

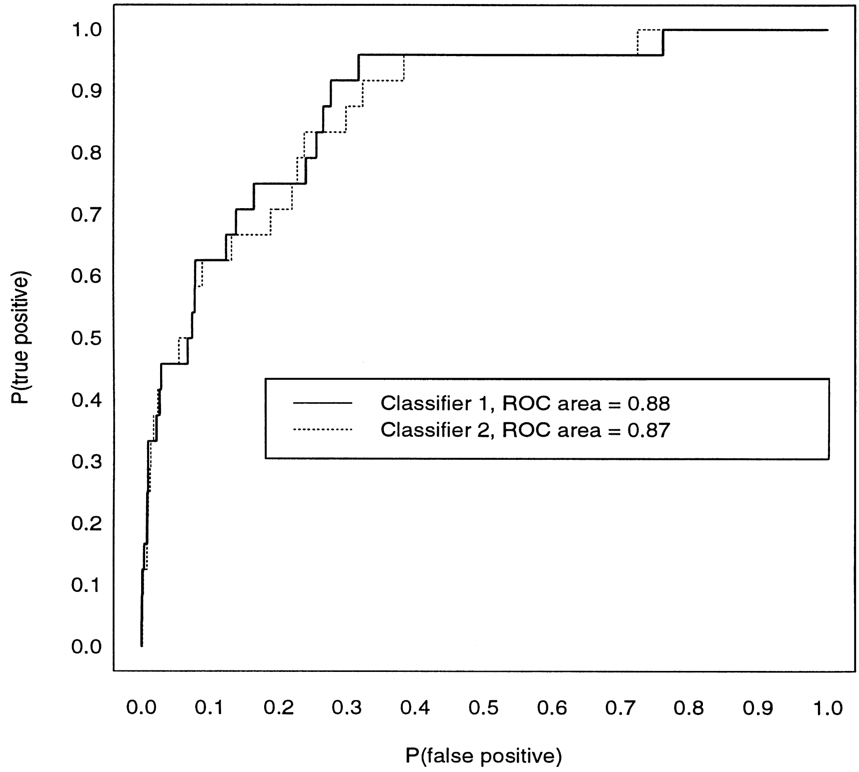


Fig. 1. Comparison between two classifiers using the areas under their ROC curves.

2.2.2. *Kullback–Leibler distance*

A natural measure of distance within the unit interval is the Kullback–Leibler (KL) distance. This distance measures the closeness between the observed y_i given $\boldsymbol{x}_i$ and the predicted $\hat{f}(\boldsymbol{x}_i)$, $\forall i$, via

$$\sum_i \left(y_i \log \frac{y_i}{\hat{f}(\boldsymbol{x}_i)} + (1 - y_i) \log \frac{1 - y_i}{1 - \hat{f}(\boldsymbol{x}_i)} \right).$$

The smallest distance is obviously 0 which happens when $\hat{f}(\boldsymbol{x}_i) = y_i$, $\forall i$.

2.2.3. *Relationship between ROC area and Kullback–Leibler distance*

Discrimination and calibration are two related yet different measures. Although a model with good discrimination tends to have good calibration and vice versa, a model may appear to be strong in one measure but weak in the other. Consider we have 200 negatives and 200 positives in a data set. Suppose the predicted probabilities from a model (say model A) are 0.20 for 100 negatives and 0.50 for the other 100 negatives, and 0.50 for 100 positives and 0.80 for the other 100 positives. The resulting ROC area is 0.875 with a Kullback–Leibler distance of 183. From another model (model B), the predicted probabilities are 0.00 for 100 negatives and 0.51 for the other 100 negatives, and 0.49 for 100 positives and 1.00 for the other 100 positives. Model B results in an inferior ROC area of 0.75 (less separability than model A) but a superior Kullback–Leibler distance of 143 (closer to the targets).

Harrell et al. (1996) recommended that good discrimination be preferred to good calibration since a model with good separability can always be recalibrated but the rank orderings of probabilities cannot be changed to improve separation. Although we computed both the ROC area and the Kullback–Leibler distance to assess model performance, we adopted the recommendation of Harrell, Lee, and Mark and used ROC area as the guiding measure when it came to the selection of the optimal amount of noise variance.

3. Simulations

We applied five classification models, specifically, 1-nn method, $q-2-1$ neural net, full-grown tree, pruned tree, and quadratic discriminant to the following four data sets: two simulated data sets and two real-data sets. Five hundred pairs of training and validation data sets (T, V) were either independently simulated from the known distributions, or randomly drawn from the real-data sets. The first simulated data involves a standard task of classifying two bivariate normals. The second simulated data is much different from the first. The training data is generated from a deterministic rule which is not known to the observer. The task is to learn the hidden rule from a finite training data set and to classify *future* unseen cases. It seems counterintuitive to introduce noisy replicates to a training data set generated from a deterministic system with no noise, but we showed that, even in this case, training with noisy replicates does result in an improved prediction. The two real-data sets concern medical diagnosis problems and are available from the Information and Computer Science repository of the University of California at Irvine. To demonstrate the effectiveness of our technique to skewed classifications, we made the problem harder by increasing the skew between the two classes.

3.1. Data sets

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

3.1.1. Bivariate normals

The training data set T consists of the following:

$$T = \left\{(\boldsymbol{x}_i, 0),\ \boldsymbol{x}_i \sim \boldsymbol{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right),\ i = 1, \ldots, 200\right\} \cup$$
$$\left\{(\boldsymbol{x}_i, 1),\ \boldsymbol{x}_i \sim \boldsymbol{N}\left(\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1 & -0.5\\-0.5 & 1\end{pmatrix}\right),\ i = 1, \ldots, 20\right\}$$

Class 0 is the numerous class which has 200 observations and class 1 is the rare class which has 20 observations. The validation data set V is independently simulated from the same underlying distributions with the same number of class 0 and 1 observations.

Table 1
Financial advising rule

(1)	If (saving-adequate and income-adequate) then *invest-stocks*
(2)	If dependent-saving-adequate then saving-adequate
(3)	If assets-high then saving-adequate
(4)	If dependent-income-adequate then income-adequate
(5)	If debt-low then income-adequate
(6)	If ($\mathit{saving} \geq \mathit{dependents}\times 5000$) then dependent-saving-adequate
(7)	If ($\mathit{income} \geq 2500 + 4000\times \mathit{dependents}$) then dependent-income-adequate
(8)	If ($\mathit{assets} \geq \mathit{income}\times 10$) then assets-high
(9)	If ($\mathit{annual\text{-}debt} < \mathit{income}\times 0.3$) then debt-low

3.1.2. Finance

The data set is generated from a deterministic rule adapted from (Luger and Stubblefield, 1989) and is shown in Table 1. It is a simplified system trying to illustrate issues involved in real-life financial advising. The system consists of five input variables, shown in bold in Table 1, and a binary response variable *invest-stocks* where 0 represents no investments in stocks and 1 represents investments in stocks. These five input variables are generated randomly according to the following — *savings*: uniform random number ranging from \$5000 to \$50,000, *dependents*: uniform random integer ranging from 1 to 6, inclusively, *income*: uniform random number ranging from \$10,000 to \$50,000, *assets*: *income* × uniform random number ranging from 0 to 20, *annual-debt*: *income* × uniform random number ranging from 0 to 0.6. The training data set consisting of 200 class 0 cases and 20 class 1 cases is randomly generated from this rule. The validation data set consists of the same number of class 0 and 1 observations.

3.1.3. 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.

3.1.4. 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 which consist 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.

3.2. Simulation algorithm

The following algorithm was used to conduct the simulation experiments. There are four pre-chosen simulation parameters in the algorithm: *noisy.repl*, *noisy.train*, *sigma.step*, and *num.sim*. They represent, in order, the number of noisy replicates generated for a given rare case, the number of noisy training data sets generated for a given training data set, the increment of the standard deviation of the noise σ_{noise}, and the number of pairs of training data set and validation data set generated in the simulation.

Step A: Initialize σ_{noise}.

Step B: Initialize t, the index denotes the tth training data set T^t and the tth validation data set V^t.

Step 1: T^t and V^t are independently drawn without replacement from a given real-data set, or are randomly simulated from a known underlying distribution.

Step 2: Models are fitted to *noisy.train* versions of noisy training data sets T_k^{t*}, $k = 1,\ldots,noisy.train$.

(a) Initialize the first version of noisy training data set. Let $k = 1$.

(b) Noisy replicates of the rare cases in $T^t = \{(\boldsymbol{x}_i, 0),\ i = 1,\ldots,n_0\} \cup \{(\boldsymbol{x}_i, 1),\ i = 1,\ldots,n_1\}$ are added, the resulting noisy training data is denoted by $T_k^{t*} = \{(\boldsymbol{x}_i, 0),\ i = 1,\ldots,n_0\} \cup \{(\boldsymbol{x}_i + \varepsilon_{ijk}, 1),\ i = 1,\ldots,n_1;\ j = 1,\ldots,noisy.repl\}$, where $\varepsilon_{ijk} \sim \boldsymbol{N}_q(\boldsymbol{0}, \sigma^2_{noise}\boldsymbol{\Sigma}_q)$, and $\boldsymbol{\Sigma}_q$ is the $q \times q$ diagonal matrix $diag\{s_1^2,\ldots,s_q^2\}$, with s_l^2 the sample variance of the lth explanatory variable x_l over the training data set.

(c) Classification models are fitted to T^t and T_k^{t*}, and the estimated models are denoted by $Model_{T^t}$ and $Model.noisy_{T_k^{t*}}$, respectively.

(d) Let Y be the vector of all the observed class labels in the validation data set V^t, i.e., $Y = \{y\colon (\boldsymbol{x}, y) \in V^t\}$. The vector Y is estimated via the two models $Model_{T^t}$ and $Model.noisy_{T_k^{t*}}$. Let $\hat{Y}$ denote the corresponding vector of conditional class label probabilities estimated via $Model_{T^t}$, i.e., $\hat{Y} = \{\hat{f}_{T^t}(\boldsymbol{x})\colon (\boldsymbol{x}, y) \in V^t\}$. Let $\hat{Y}_k^*$ denote the corresponding vector of conditional class label probabilities estimated via $Model.noisy_{T_k^{t*}}$, i.e., $\hat{Y}_k^* = \{\hat{f}_{T_k^{t*}}(\boldsymbol{x})\colon (\boldsymbol{x}, y) \in V^t\}$. Note that we suppressed the index t in $Y, \hat{Y}$, and $\hat{Y}_k^*$ for notational simplicity.

(e) If $k < noisy.train$, then $k = k + 1$, and go back to (a). Otherwise, continue to Step 3.

Step 3: Average the *noisy.train* vectors $\hat{Y}_k^*$, $k = 1,\ldots,noisy.train$, to obtain a vector of conditional class label probabilities $\hat{Y}^*$ estimated via the *noisy.train* noisy training data sets. That is,

$$\hat{Y}^* = \frac{\sum_{k=1}^{noisy.train} \hat{Y}_k^*}{noisy.train}.$$

Table 2
Summary of the pilot study on the mean ROC area

Model	Original ROC area	*noisy.repl*	*noisy.train*				
			2	4	6	8	10
1-nn	0.585	1	0.607	0.651	0.684	0.701	0.709
		2	0.692	0.728	0.744	0.751	0.769
		3	0.710	0.751	0.766	0.774	0.779
$q-2-1$ NN	0.805	1	0.825	0.842	0.854	0.857	0.858
		2	0.855	0.863	0.862	0.863	0.864
		3	0.859	0.860	0.863	0.862	0.863
Full-grown tree	0.695	1	0.738	0.752	0.787	0.810	0.816
		2	0.759	0.796	0.807	0.818	0.823
		3	0.754	0.803	0.819	0.822	0.830
Pruned tree	0.709	1	0.758	0.786	0.816	0.825	0.825
		2	0.775	0.818	0.824	0.835	0.841
		3	0.779	0.819	0.832	0.839	0.839
QD	0.867	1	0.861	0.862	0.863	0.864	0.863
		2	0.864	0.864	0.863	0.863	0.864
		3	0.862	0.864	0.864	0.864	0.864

The quality of the two sets of predictions $\hat{Y}$ and $\hat{Y}^*$ is summarized by the ROC areas ROC^t and ROC^{t*}, and the Kullback–Leibler distances KL^t and KL^{t*}, respectively.

Step C: Let *num.sim* represent the number of times we repeat the experiment. If $t < num.sim$, then $t = t + 1$, and go to Step 1. If $t = num.sim$ and the average of $\{ROC^{t*} - ROC^t,\ t = 1,\dots,num.sim\}$ remains positive (which means that the noisy replicates are still beneficial in the average), then increase σ_{noise} to $\sigma_{noise} + sigma.step$, set $t = 1$, and repeat Step B. Otherwise, stop.

We ran a pilot study on the Bivariate Normals data set to test out the different behavior of the simulation parameters *noisy.repl*, *noisy.train*, *sigma.step*, and *num.sim*. We tried all combinations of $noisy.repl = 1, 2, 3$ and $noisy.train = 2, 4, 6, 8, 10$, while keeping $sigma.step = 0.5$ and $num.sim = 20$. Tables 2 and 3 summarize the results of the pilot study for the five classification models: 1-nn, $q-2-1$ neural net, full-grown tree, pruned tree, and quadratic discriminant. The following facts were observed. The smallest non-trivial *noisy.repl*, which is two, would be enough to show improvement. We want to average at least several (the more the better) estimates from different noisy training data sets generated for a given training data. The increment $sigma.step = 0.5$ is sufficient to produce some positive results. The number of pairs of training data set and validation data set generated is not a problem as long as it is reasonably large, the larger the better. As a result of balancing computing time and simulation details, we chose $noisy.repl = 2$, $noisy.train = 10$, $sigma.step = 0.5$, and $num.sim = 500$ for all data sets and models (Tables 4 and 5). All classification

Table 3
Summary of the pilot study on the mean KL distance

Model	Original	*noisy.repl*	*noisy.train*				
	KL distance		2	4	6	8	10
1-nn	6608	1	3988	2787	2193	1852	1658
		2	4196	2538	1968	1592	1227
		3	5997	3412	2670	2214	1975
$q-2-1$ NN	678	1	66	53	52	52	52
		2	150	65	61	60	60
		3	78	77	75	75	75
Full-grown tree	2149	1	1363	953	464	259	214
		2	1162	456	275	161	127
		3	1139	590	429	327	258
Pruned tree	462	1	155	60	57	57	56
		2	130	82	59	59	59
		3	92	67	65	66	65
QD	52	1	52	52	52	52	52
		2	57	58	58	58	58
		3	66	66	66	66	66

models are either standard Splus functions or are available in the extended Splus library in *Statlib*. The simulations were performed on a HP-UNIX platform using Splus version 3.4.

4. Results

For $t=1,\ldots,500$, define $\Delta ROC^t = ROC^{t*} - ROC^t$, where ROC^t and ROC^{t*} denote the respective ROC areas for the tth original and noisy model predictions. When ΔROC^t is positive, the noisy model is better because its predictions $\hat{Y}^*$ are more "separable" than the original model predictions $\hat{Y}$. Based on $\{\Delta ROC^t,\ t=1,\ldots,500\}$, we computed a 95% confidence interval estimate for the unknown population means $\mu_{\Delta ROC}$, where $\mu_{\Delta ROC}$ is the true mean change in the ROC area when noisy replicates are introduced during training. To be precise, the mean change in ROC area $\mu_{\Delta ROC}$ should be written as $\mu_{\Delta ROC,\sigma_{noise}}$ since the change in ROC area is a function of σ_{noise}. We started with the smallest noise standard deviation at 0.001 instead of exactly at 0 to allow for non-identical noisy replicates.

If there is no significant difference, at 5% type I error, in adding noisy replicates during training, then the 95% confidence interval will contain 0. If the addition of noisy replicates during training improves the prediction, then the 95% confidence interval should not contain 0 and the entire interval should be positive (i.e., above

Table 4
Summary of the 500 ROC areas for the original and optimal noisy models for the four data sets

Data	Model	Opt. σ_{noise}	Mean ROC area[a] *Orig* vs. *Noisy*	% increase
Normals	1-nn	0.5	$0.603 < 0.771$	27.9%
	$q-2-1$ NN	0.5	$0.812 < 0.869$	6.9%
	Full-grown tree	0.5	$0.709 < 0.838$	18.2%
	Pruned tree	0.5	$0.735 < 0.849$	15.4%
	QD	0.5	$0.869 < 0.873$	0.6%
Finance	1-nn	1.0	$0.632 < 0.788$	24.7%
	$q-2-1$ NN	0.5	$0.756 < 0.849$	12.3%
	Full-grown tree	0.5	$0.704 < 0.840$	19.4%
	Pruned tree	0.5	$0.671 < 0.815$	21.4%
	QD	1.0	$0.733 < 0.804$	9.8%
Diabetes	1-nn	1.5	$0.544 < 0.609$	11.9%
	$q-2-1$ NN	0.5	$0.676 < 0.779$	15.1%
	Full-grown tree	1.0	$0.621 < 0.735$	18.3%
	Pruned tree	1.0	$0.672 < 0.755$	12.4%
	QD	1.0	$0.629 < 0.687$	9.2%
Hypothyroid	1-nn	1.0	$0.712 < 0.861$	20.9%
	$q-2-1$ NN	0.5	$0.867 < 0.958$	10.5%
	Full-grown tree	1.0	$0.873 < 0.932$	6.7%
	Pruned tree	1.5	$0.903 < 0.942$	4.4%
	QD	0.5	$0.898 < 0.903$	0.5%

[a] $a < b$ indicates that the mean ROC area a is significantly smaller than the mean ROC area b with p-value less than 0.01.

the x-axis), and vice versa. The confidence intervals computed for all the five levels of $\sigma_{noise} = 0.001, 0.5, 1.0, 1.5, 2.0$ were plotted in Figs. 2–5.

Each figure consists of five plots which correspond to the five classification models, 1-nn, $q-2-1$ neural net, full-grown tree, pruned tree, and quadratic discriminant, fitted to each data set. Each plot consists of two lines connecting the five confidence intervals obtained according to the five levels of σ_{noise}. The solid line connects the five confidence intervals obtained with averaging when $noisy.train = 10$, and the dashed line connects the five confidence intervals obtained with no averaging, i.e., when $noisy.train = 1$. The main result obtained by averaging multiple model estimates is represented by the solid line; the dashed line is drawn as a baseline reference to illustrate the substantial gain achieved by averaging. The solid line is above the dashed line, showing that averaging does improve the results. The improvement is significant when the 95% confidence intervals are disjoint. In other words, at those levels of σ_{noise} with disjoint confidence intervals, the estimate, $\hat{f}_{\bar{T}^*}(\boldsymbol{x})$, is found to be significantly better than $\hat{f}_{T^*}(\boldsymbol{x})$.

It is clear from these figures that most of the confidence intervals joined by the solid line are positive and lie well above the x-axis, indicating the existence of an

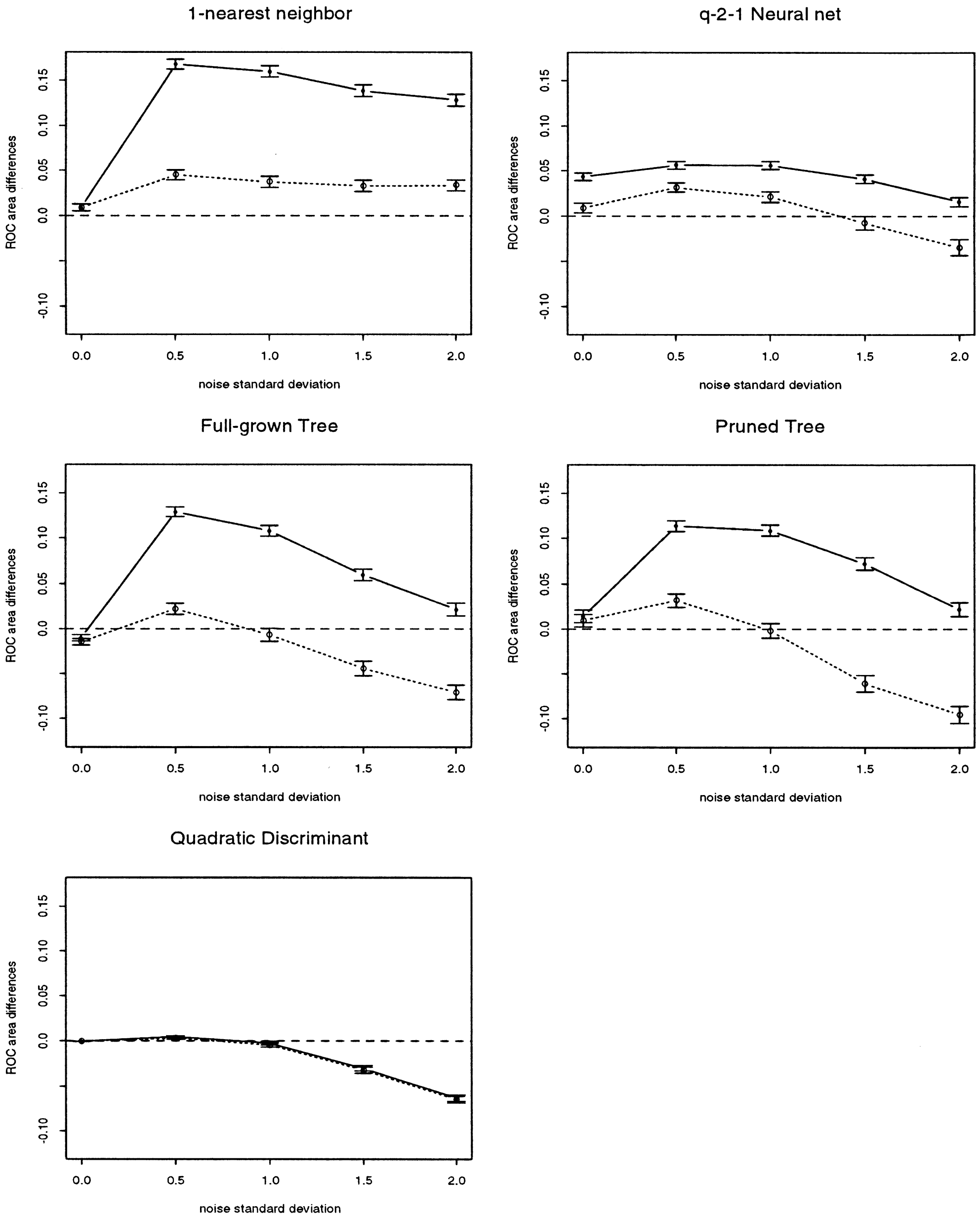


Fig. 2. 95% Confidence Intervals of the ROC area differences for the Normals Data. (Solid line is for averaging; dashed line is for no averaging).

optimal σ_{noise} which could maximize the significant improvement. We start to see some positive results even at $\sigma_{noise} = 0.001$ for some models, especially for neural nets. It is interesting to observe the jump increase from σ_{noise}= 0.001–0.5 in many models. As expected, the performance eventually drops for σ_{noise} larger than 2.0

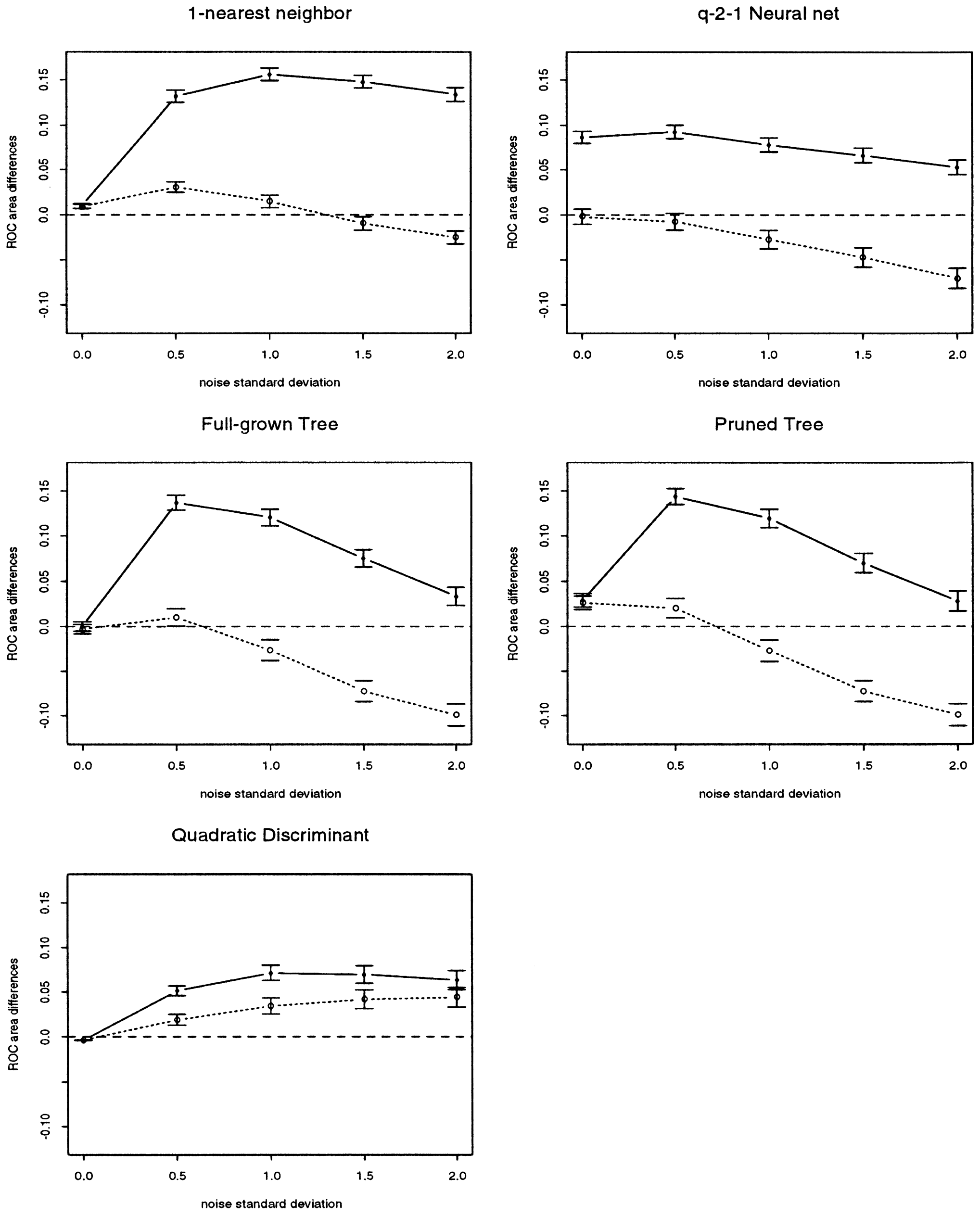


Fig. 3. 95% Confidence Intervals of the ROC area differences for the Finance Data. (Solid line is for averaging; dashed line is for no averaging).

because further increase in the noise standard deviation will seriously corrupt the original training data set. Optimal value of σ_{noise} is selected when the ROC area reaches its peak performance. That is to say, $\mu_{\Delta ROC, optimal \sigma_{noise}}$ is the highest of the five $\sigma_{noise} = 0.001, 0.5, 1.0, 1.5, 2.0$. At this optimal σ_{noise}, the estimate, $\hat{f}_{\bar{T}^*}(\boldsymbol{x})$, would

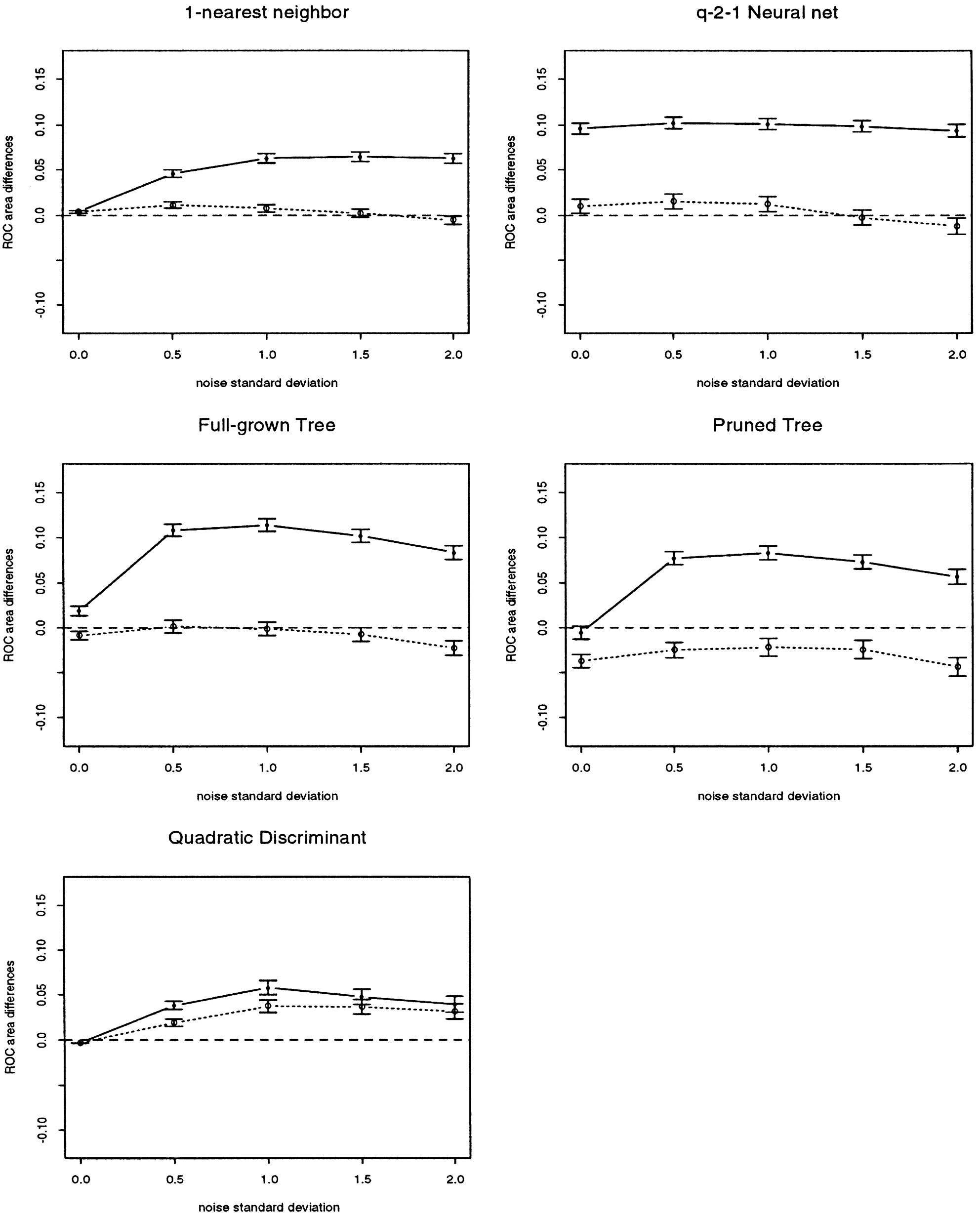
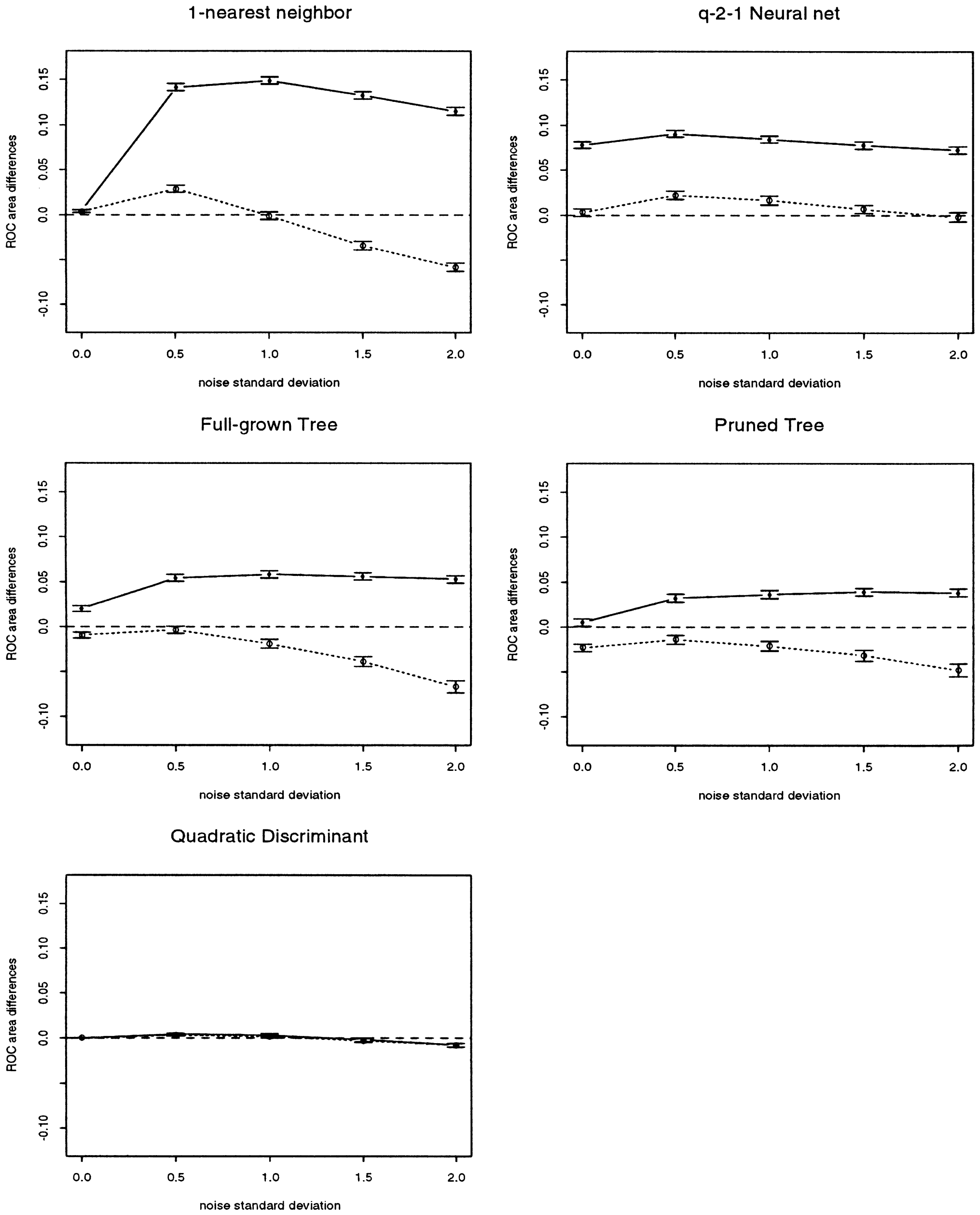


Fig. 4. 95% Confidence Intervals of the ROC area differences for the Diabetes Data. (Solid line is for averaging; dashed line is for no averaging).

be significantly better than the original (no noisy replicates) estimate $\hat{f}_T(\boldsymbol{x})$ when the corresponding confidence interval stays above the x-axis.

To further summarize our findings when the optimal σ_{noise} was chosen, we tabulate the mean ROC areas and the mean Kullback–Leibler distances for all the data sets and models combinations in Tables 4 and 5. In Table 4, the label Opt. σ_{noise}



1-nearest neighbor

q-2-1 Neural net

Full-grown Tree

Pruned Tree

Quadratic Discriminant

Fig. 5. 95% Confidence Intervals of the ROC area differences for the Hypothyroid Data. (Solid line is for averaging; dashed line is for no averaging).

represents the selected optimal σ_{noise} level. The label *Orig* represents the original model without noisy replicates, and *Noisy* represents the optimal noisy model with optimal σ_{noise} and with $noisy.train = 10$. Paired t-tests were conducted to compare the ROC area and Kullback–Leibler distance for the optimal noisy model versus

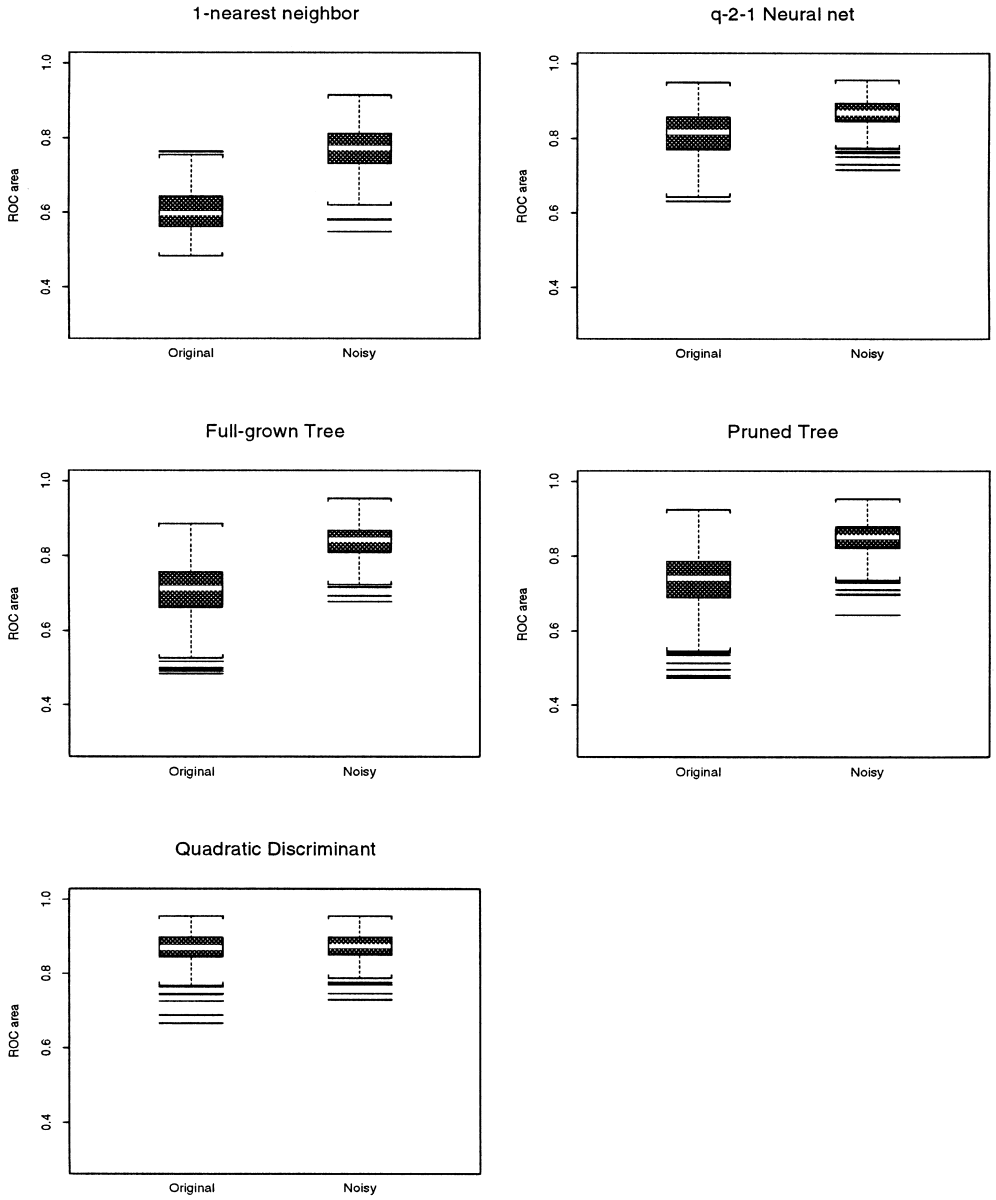
Table 5
Summary of the 500 KL distances for the original and optimal noisy models for the four data sets.

Data	Model	Mean KL distance[b] *Orig* vs. *Noisy*	% decrease
Normals	1-nn	$6570 > 1599$	76%
	$q-2-1$ NN	$485 > 57$	88%
	Full-grown tree	$2093 > 203$	90%
	Pruned tree	$463 > 57$	88%
	QD	$50 < 53$	−5%
Finance	1-nn	$2287 > 617$	73%
	$q-2-1$ NN	$877 > 44$	95%
	Full-grown tree	$1181 > 174$	85%
	Pruned tree	$520 > 41$	92%
	QD	$64 > 32$	50%
Diabetes	1-nn	$5445 > 2383$	56%
	$q-2-1$ NN	$1550 > 85$	95%
	Full-grown tree	$2401 > 704$	71%
	Pruned tree	$863 > 117$	86%
	QD	$177 > 101$	43%
Hypothyroid	1-nn	$6126 > 2146$	65%
	$q-2-1$ NN	$976 > 108$	89%
	Full-grown tree	$2596 > 970$	63%
	Pruned tree	$1254 > 448$	64%
	QD	$152 > 138$	9%

[b] $a < b$ indicates that the mean Kullback–Leibler distance a is significantly smaller than the mean Kullback–Leibler distance b, $a > b$ indicates that the mean Kullback–Leibler distance a is significantly larger than the mean Kullback–Leibler distance b, both with p-value less than 0.01.

the original model. All ROC areas obtained from optimal noisy models are significantly larger than that from original models with p-values less than 0.01. All, except one, Kullback–Leibler distances obtained from optimal noisy models are significantly smaller than that from original models with p-values less than 0.01. The percentage change (increase or decrease) is calculated via $(Noisy - Orig)/Orig \times 100\%$. All optimal noisy models have superior (increased) ROC areas with percentage increases ranging from 0.5% to 27.9% and most of them exceed 10%. All, except one, have superior (decreased) Kullback–Leibler distances with impressive percentage decreases, most of them are larger than 50%. The only exception is the Quadratic discriminant method in the *Normals* data set. The noisy model still produces a slightly superior (0.6% increase) ROC area, but an inferior (5% increase) Kullback–Leibler distance. We would definitely prefer the original Quadratic discriminant method to separate two normals with different covariance matrices; this is an expected exception in this case!

We also provided boxplots for the 500 ROC areas and Kullback–Leibler distances, Figs. 6–13, when optimal noisy replicates were introduced. From these figures, it is clear that the optimal noisy model predictions are better than the original model



1-nearest neighbor

q-2-1 Neural net

Full-grown Tree

Pruned Tree

Quadratic Discriminant

Fig. 6. Boxplots for the 500 ROC areas obtained from the Original and the optimal Noisy models for the Normals Data.

predictions since the former are more “separable” (larger ROC area) and are “closer” to the target values (smaller Kullback–Leibler distance). It is interesting to observe that the optimal σ_{noise} chosen via ROC area also produce better Kullback–Leibler distance.

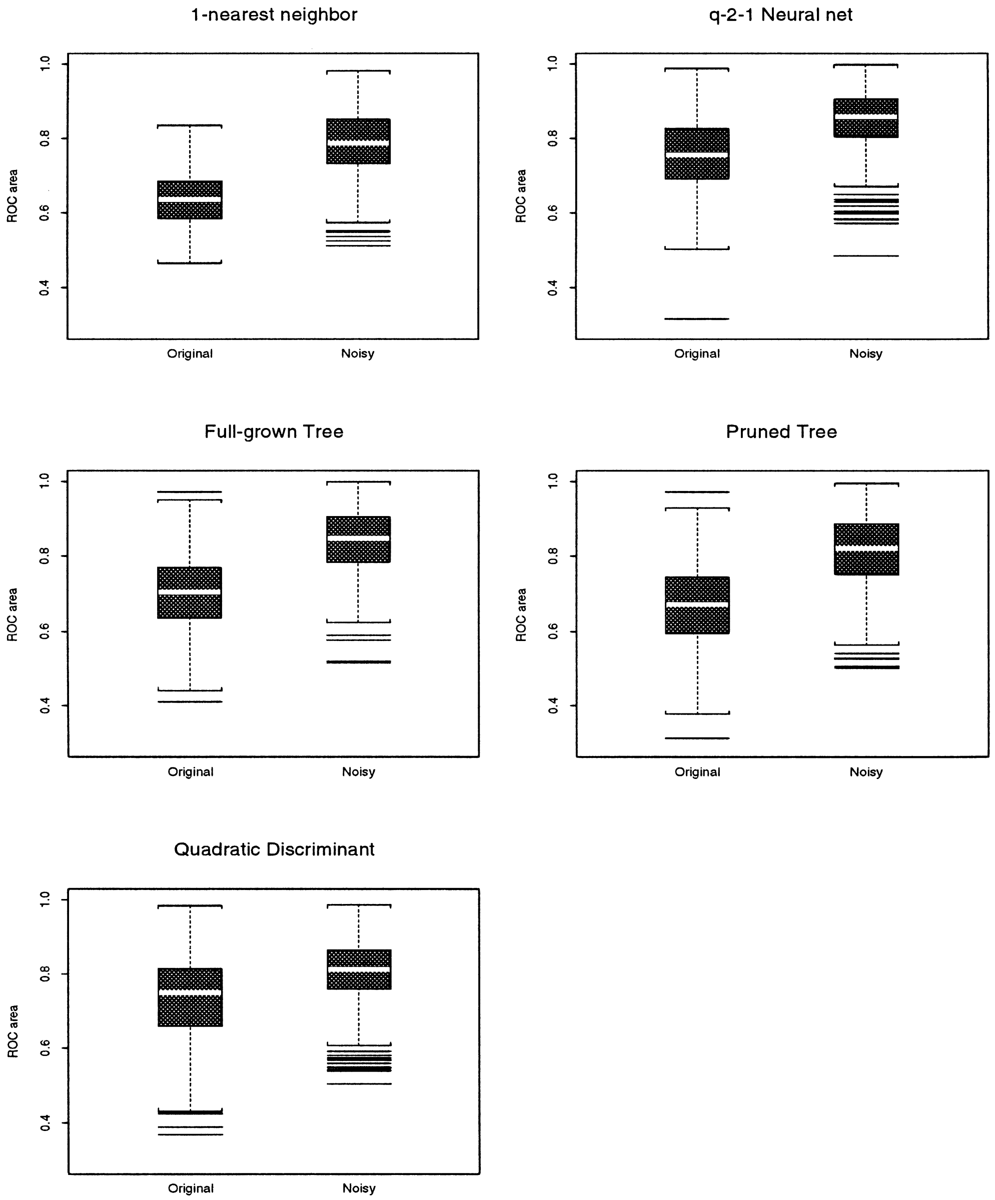


Fig. 7. Boxplots for the 500 ROC areas obtained from the Original and the optimal Noisy models for the Finance Data.

5. Discussion

The success of adding noisy replicates to the rare cases during training is demonstrated for the 1-nn method, $q-2-1$ neural net, full-grown tree, pruned tree, and quadratic discriminant in the context of skewed binary classification. The improvement could be explained by the trade off between bias and variance of the estimates

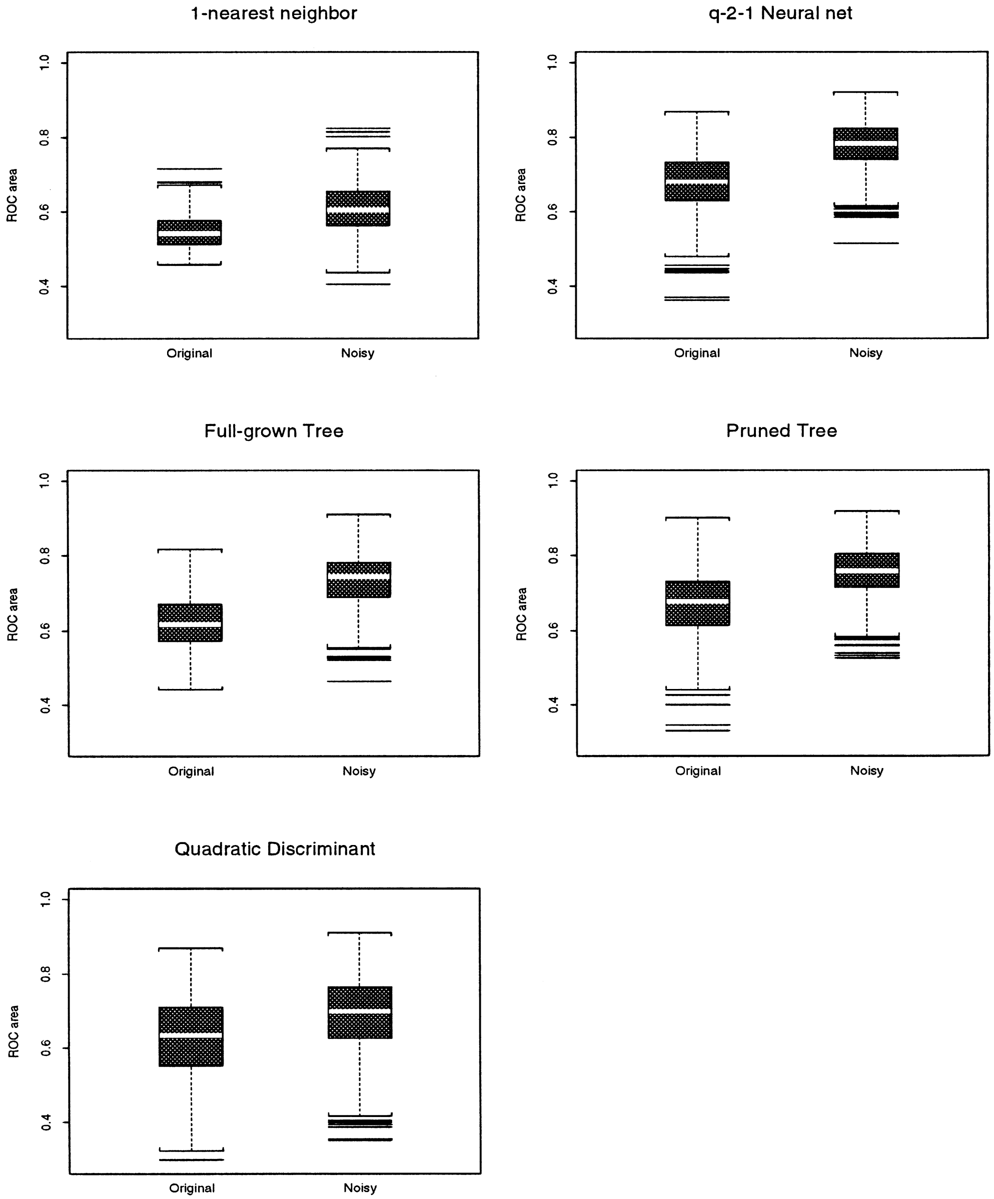


Fig. 8. Boxplots for the 500 ROC areas obtained from the Original and the optimal Noisy models for the Diabetes Data.

in the process of averaging biased and relatively independent model estimates. It is clear that adding noise will increase the bias of an estimator. However, the variance of the estimate can be drastically reduced by averaging the estimates over several noisy training data sets due to their relative independence. We have shown that, for the skewed classification considered, the net gain in the overall performance is impressive. It results in a larger ROC area and a smaller Kullback–Leibler distance.

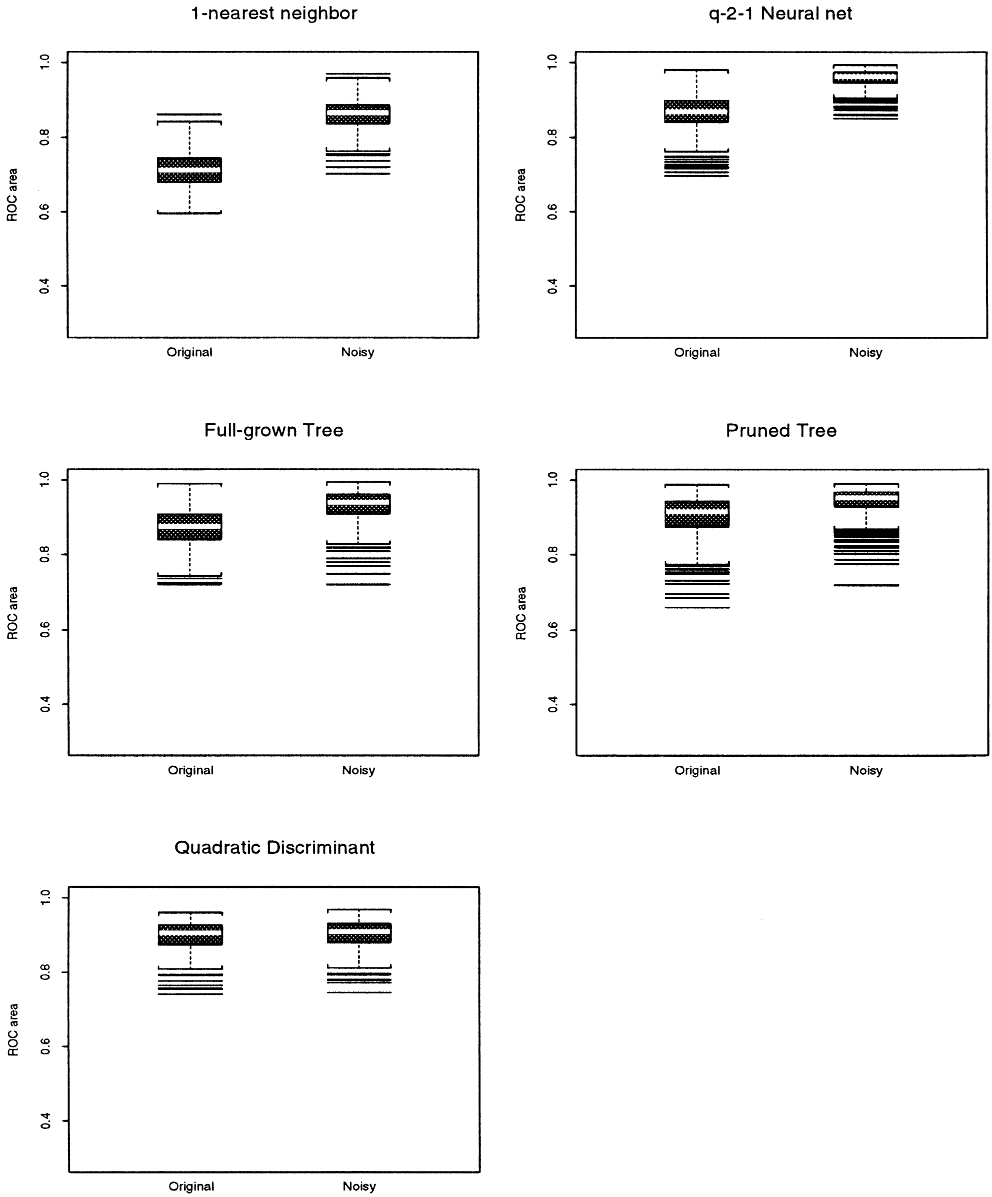


Fig. 9. Boxplots for the 500 ROC areas obtained from the Original and the optimal Noisy models for the Hypothyroid Data.

The success lies in the fact that averaging the estimates from different noisy training sets help to regularize the models.

Many forms of regularization exist for different models and they are model-specific. For example, neural networks could be regularized via weight decay, classification trees could be regularized through pruning (Venables and Ripley, 1994), and many regularization techniques exist for quadratic discriminant (Mkhadri et al., 1997). In

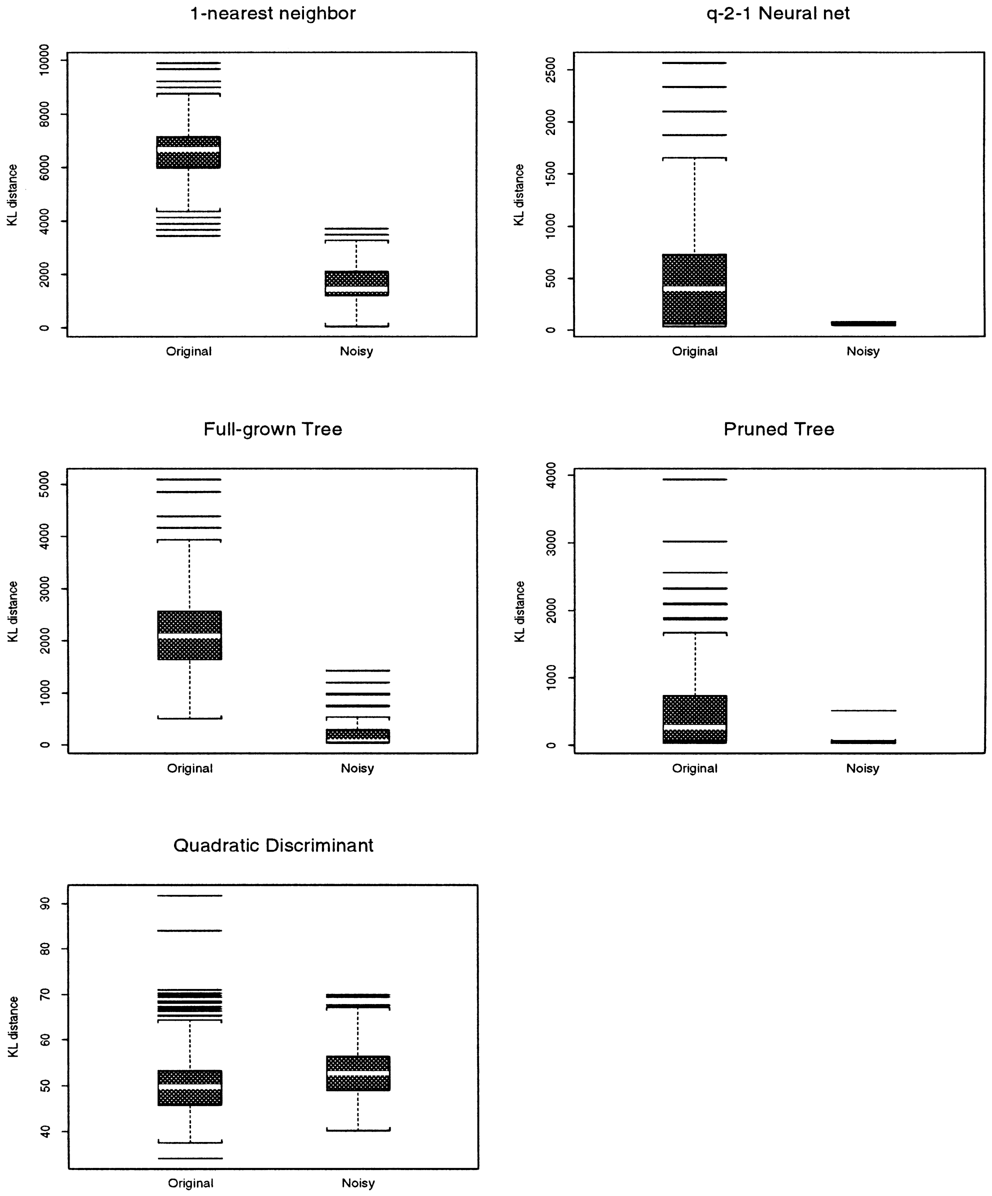


Fig. 10. Boxplots for the 500 KL distances obtained from the Original and the optimal Noisy models for the Normals Data.

this paper, positive results were obtained even in the case of pruned trees, suggesting that the effect of noisy replicates is complementary to pruning — a model-specific regularization. The regularization by noisy replicates is conceptually different from the model-specific regularization. For the classification models considered, adding noisy replicates to the rare cases in skewed binary classification offers a simple, elegant, and unified treatment for regularization which is model-free.

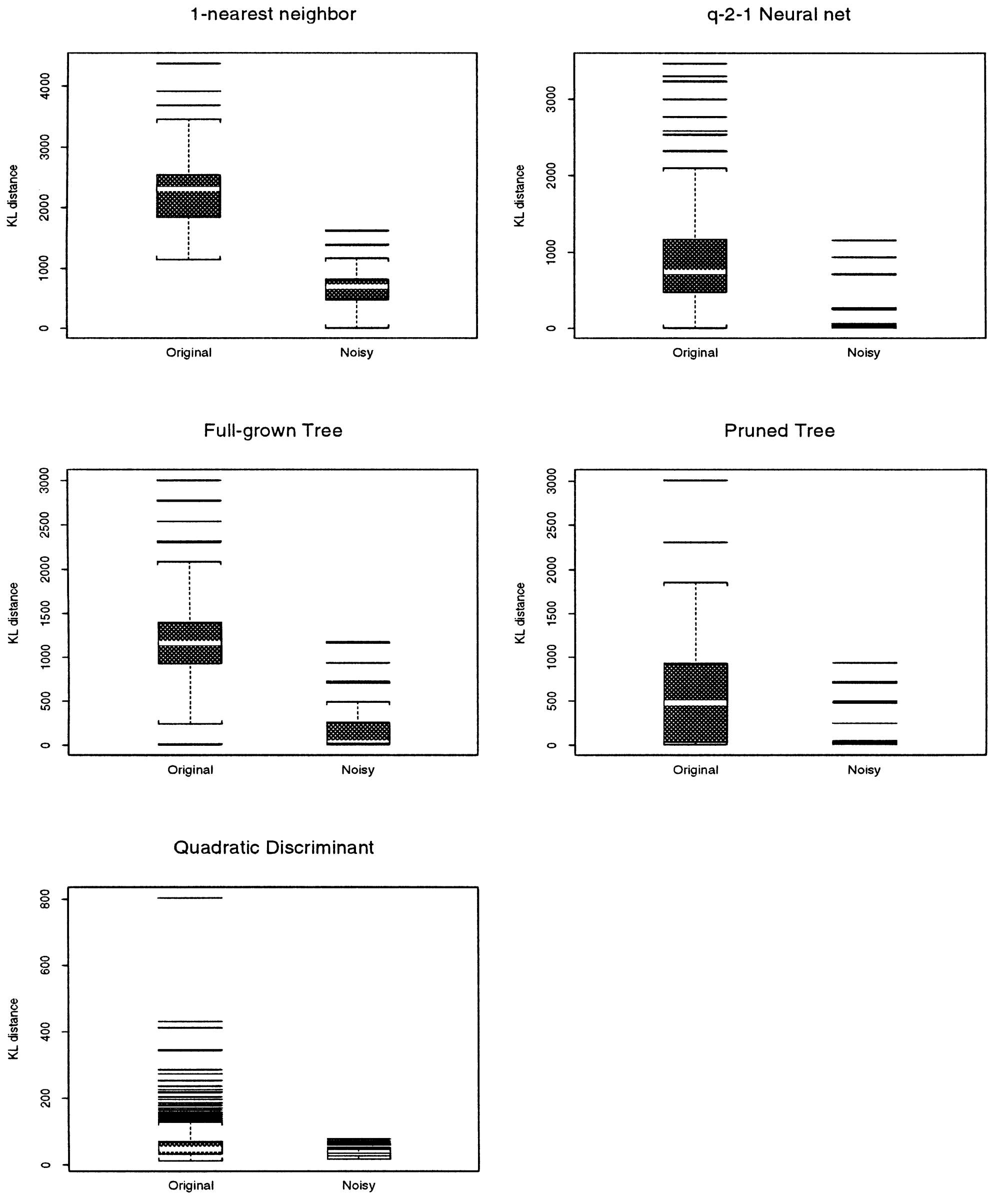


Fig. 11. Boxplots for the 500 KL distances obtained from the Original and the optimal Noisy models for the Finance Data.

6. Concluding remark

Note that we uniformly chose *noisy.repl* = 2, *noisy.train* = 10, *sigma.step* = 0.5, and *num.sim* = 500 for all classification models and all data sets to demonstrate that the averaging of noisy replicates is superior. Better result could be obtained when the aim is to find an optimal classification of a particular data set. In that case,

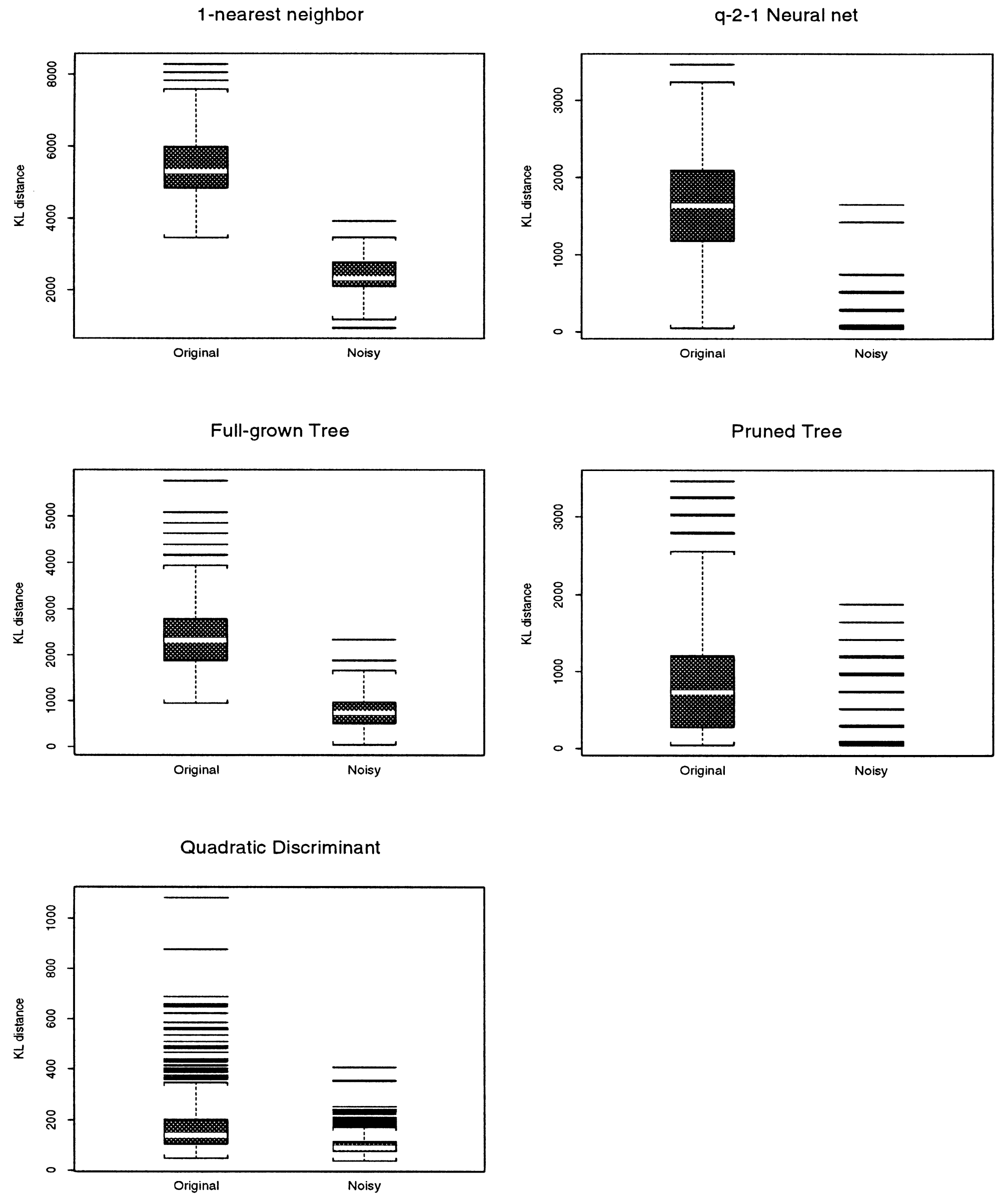


Fig. 12. Boxplots for the 500 KL distances obtained from the Original and the optimal Noisy models for the Diabetes Data.

the choice of *noisy.repl*, *noisy.train*, and σ_{noise} (a multiple of *sigma.step*) can be uniquely determined by resampling through cross validation. The idea is to gradually increase the value of the parameters *noisy.repl*, *noisy.train*, and σ_{noise} until the cross validation ROC area begins to decrease. The optimal values of these parameters are located at the turning point where the cross validation ROC area changes from increasing to decreasing.

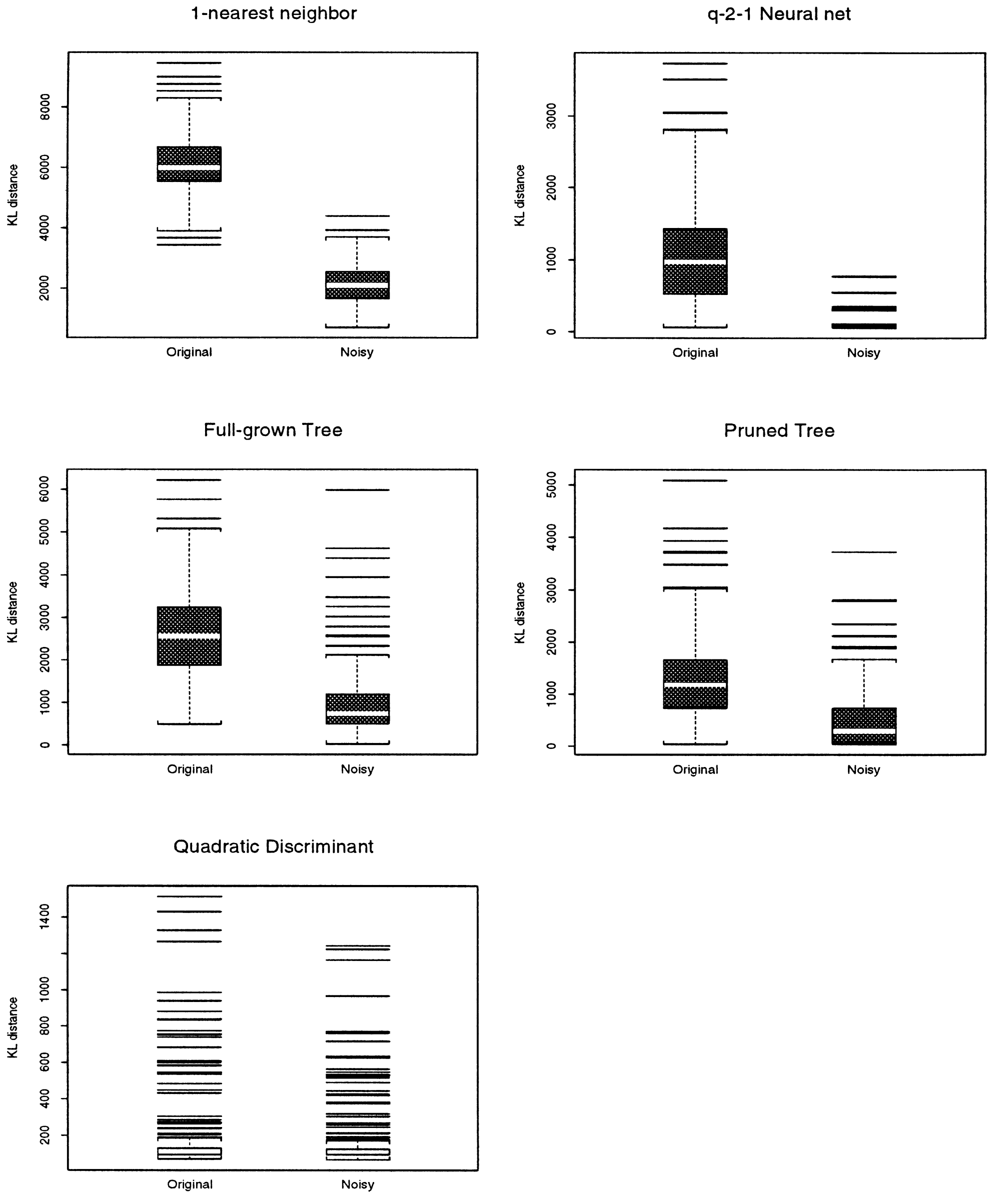


Fig. 13. Boxplots for the 500 KL distances obtained from the Original and the optimal Noisy models for the Hypothyroid Data.

This technique of averaging noisy replicates, however, is not recommended for relatively simple models such as linear discriminant and logistic regression. To demonstrate this point, we performed simulation (with *noisy.repl* = 2 and *num.sim* = 500 across all the five levels of σ_{noise}) on both of these two models and observed that noisy models slightly degraded the original performance. Due to the simplistic nature of the two models, the introduction of the noisy replicates did not change $\hat{f}_{T^*}(\boldsymbol{x})$

much and produced only a slight bias. Subsequently, averaging similar values of $\hat{f}_{T^*}(\boldsymbol{x})$ from different noisy training data sets did not yield any improvement over $\hat{f}_T(\boldsymbol{x})$. We tried averaging (*noisy.train* = 10) as well as no averaging (*noisy.train* = 1) for these two models and the results were not much different. The models seemed to produce very stable estimates of $\hat{f}_{T^*}(\boldsymbol{x})$ for various T^* and thus this regularization technique was unnecessary for linear discriminant and logistic regression.

It is not clear to us why the improvement for the quadratic discriminant is least generally. Based on what we have observed so far, we conjecture that parametric models will generally produce more stable estimates of $\hat{f}_{T^*}(\boldsymbol{x})$ for various T^*. As a consequence, the improvement for QD is not as impressive as the other nonparametric methods.

To conclude, we demonstrated that adding noisy replicates to skewed binary classification is a successful and natural form of regularization for the classification models considered. The improvements were shown to be statistically significant with p-values less than 0.01 and practically significant with impressive percentage gains in terms of ROC area and Kullback–Leibler distance. It is our hope that the success for these models studied in this paper will provide a basis of extending the technique to other models, including k nearest-neighbor method with $k > 1$ and neural networks with more than 1 hidden layer or more than two hidden units in a single hidden layer. It is observed that the classification models considered in this paper have these characteristics: highly local and case-based (1 nearest neighbor method), highly nonparametric and flexible ($q - 2 - 1$ neural networks, classification trees), or low cases to parameters ratio (rare cases to μ_1 and $\boldsymbol{\Sigma}_1$ in QD). An open research question is: for models with the above-mentioned characteristics (e.g., general additive model GAM and projection pursuit regression), how effective is the noisy replication technique to skewed binary classification? Other potential areas include fine tuning of the various simulation parameters *noisy.repl*, *noisy.train*, *sigma.step*, and relaxing the assumption of the variance covariance matrix of the noise $\boldsymbol{\Sigma}_q$ being diagonal. Our immediate interest is to see how the technique performs when applied to general classification problems which are not necessarily skewed or binary.

References

Breiman, L., 1996. Bagging predictors. Mach. Learning 26 (2), 123–140.

Breiman, L., 1998. Arcing classifiers. Ann. Statist. 26, 801–849.

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

Freund, Y., Schapire, R., 1996. Experiments with a new boosting algorithm. Machine Learning: Proceedings of the 13th International Conference, pp. 148–156.

Hand, D.J., 1997. Construction and Assessment of Classification Rules. Wiley, New York.

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

Harrell, F.E., Lee, K.L., Mark, D.B., 1996. Multivariate prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Statist. Med. 15, 361–387.

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

Lee, S.S., 1999. Regularization in skewed binary classification. Comput. Statist. 14, 277–292.

Luger, G.F., Stubblefield, W.A., 1989. Artificial Intelligence and the Design of Expert Systems Benjamin/Cummings, Menlo Park, CA.

Mkhadri, A., Celeux, G., Nasroallah, A., 1997. Regularization in discriminant analysis: an overview. Comput. Statist. Data Anal. 23, 403–423.

Raviv, Y., Intrator, N., 1996. Bootstrapping with noise: an effective regularization technique. Connection Sci., Special issue on Combining Estimators 8, 356–372.

Ripley, B.D., 1996. Pattern Recognition and Neural Networks. Cambridge University Press, UK.

Venables, W.N., Ripley, B.D., 1994. Modern Applied Statistics with S-plus. Springer, New York.