2000-07-01

# Neural networks versus nonparametric neighbor-based classifiers for semisupervised classification of Landsat Thematic Mapper imagery

Perry J. Hardin
perry_hardin@byu.edu



**Original Publication Citation**
Hardin, Perry J. "Neural networks versus nonparametric neighbor-based classifiers for semisupervised classification of Landsat Thematic Mapper imagery." Optical Engineering 39.7 (2): 1898-198

# Neural networks versus nonparametric neighbor-based classifiers for semisupervised classification of Landsat Thematic Mapper imagery

**Perry J. Hardin**
Brigham Young University
Department of Geography
676 SWKT
Provo, Utah 84602
E-mail: perry_hardin@byu.edu

**Abstract.** Semisupervised classification is one approach to converting multiband optical and infrared imagery into landcover maps. First, a sample of image pixels is extracted and clustered into several classes. The analyst next combines the clusters by hand to create a smaller set of groups that correspond to a useful landcover classification. The remaining image pixels are then assigned to one of the aggregated cluster groups by use of a per-pixel classifier. Since the cluster aggregation process frequently creates groups with multivariate shapes ill suited for parametric classifiers, there has been renewed interest in nonparametric methods for the task. This research reports the results of an experiment conducted on six Landsat Thematic Mapper images to compare the accuracy of pixel assignment performed by four nearest neighbor classifiers and two neural network paradigms in a semisupervised context. In all the experiments, both the neighbor-based classifiers and the neural networks assigned pixels with higher accuracy than the maximum-likelihood approach. There was little substantive difference in accuracy among the neighborhood-based classifiers, but the feedforward network was significantly superior to the probabilistic neural network. The feedforward network classifier generally produced the highest accuracy on all six of the images, but it was not significantly better than the accuracy produced by the best neighbor-based classifier.  [S0091-3286(00)03807-1]

Subject terms: image classification; neural networks; image processing; nonparametric classification.

Paper 990302 received Aug. 2, 1999; revised manuscript received Jan. 7, 2000; accepted for publication Jan. 14, 2000.

## 1 Introduction

In remote sensing, it is common to describe image classification workflow as either supervised or unsupervised. In supervised classification, a set of *a priori* landcover classes is initially defined. Representative geographic areas for each of the landcover classes are then identified on the ground and matched with their locations on the imagery. After the image pixels corresponding to these representative areas have been extracted from the image and statistically summarized, a classifier is trained and then used to force the remaining image pixels into one of the *a priori* classes.

A common difficulty with supervised classification is the spectral confusion between the desirable landcover classes. To ameliorate this difficulty, the usual alternative is an unsupervised methodology. In this instance, clustering is employed to place each image pixel in a spectral class without regard to a preexisting landcover scheme. The concern is to generate clusters with spectral uniformity within themselves while concurrently maintaining spectral distinctness from the other clusters. After completing the clustering phase, the analyst then tries to coax the landcover meaning from each of the cluster groups. Problems arise when the spectral groupings do not correspond to landcover classes that are intuitive or useful for the application at hand.

### 1.1 *Semisupervised Classification*

In practice, supervised classification has the advantage of an *a priori* landcover schema, whereas unsupervised classification has the surety of spectral uniqueness among classes. In an attempt to maximize the advantages of both approaches, supervised and unsupervised classification can be combined into a hybrid approach.[1] While many variations are possible, one semisupervised method can be reduced to the following general steps:

1. A set of pixels is randomly (or systematically) selected from the image without regard to any *a priori* landcover classification scheme. The size of the pixel set must be large enough to represent the variation in the scene while concurrently remaining small enough to explore statistically without exhausting the analyst's patience.
2. The small pixel set is submitted to cluster analysis. The number of groups retained in the clustering far

 0091-3286/2000/$15.00 

exceeds the number of final map landcover classes anticipated.

3. The clusters are graphed, mapped, analyzed, and then combined by an analyst in order to generate useful landcover class groupings that have minimal spectral overlap. This is the step designed to maximize the relative advantages of supervised and unsupervised classification.
4. The final pixel groupings are then treated as though they were bona fide training classes in a supervised classification scheme. A classifier is chosen, trained using the spectral signature of the groupings, and then applied to the entire satellite image. This places each of the image pixels in one of the landcover classes.

### 1.2 *New Pixel Assignment Approaches*

Semisupervised classification, like its supervised parent, has a pixel assignment step—all the pixels in the image are placed in a landcover class predicated on the signatures in a training set. Several well-known algorithms for performing pixel assignment are available, but the workhorse for the task is the maximum likelihood (ML) classifier. (The ML classifier used in this research was Fischer's linear discriminant function. Prior probabilities were proportional to the actual class membership.) In contrast to this ubiquitous parametric classifier, nonparametric classifiers employing spectral neighborhoods (e.g., first-nearest-neighbor classifier) are not commonly used. Perhaps this lack of popularity is historical—their slow execution on early image-processing computers made them prohibitively expensive for classifying large images. However, recent advances in searching algorithms,[2–4] combined with increased computer speed, allow practitioners to again consider these nonparametric algorithms for the pixel assignment phase of supervised and semisupervised classification. Applied to some images, these neighbor-based classifiers have produced higher classification accuracy than the ML method,[1] whereas other attempts to use neighborhood-based classifiers have been disappointing.[5]

The use of neural networks has also been proposed as a replacement for the ML classifier. Representative projects that contain a comparison of neural network classification with ML results include those by Benediktsson et al.,[6] Foody et al.,[7] Hepner et al.,[8] Paola and Schowengerdt,[9] German and Gahegan,[10] and Chettri et al.[11] As illustrated in these projects, and as reviewed by Atkinson and Tatnall,[12] the feedforward backpropagation multilayer perceptron network (hereafter called the *feedforward network*) is the most commonly used network model for remote-sensing classification. With few exceptions, the consensus has been that feedforward networks provide higher classification accuracy than traditional parametric ML methods, but require extensive experimentation, training time, and adequate design to use effectively.[9] Because the effectiveness of feedforward networks is so promising, substantial research effort continues to be devoted to (1) limiting heuristic approaches to network design,[9,10,13–15] (2) minimizing training time[10,16] or data volume,[14] and (3) developing simpler or more effective network paradigms.[17–19]

Although neural networks and neighbor-based classification are considered distinct methods, there are practical coincidences between some models that suggest their common conceptual underpinnings.[20] For example, an ill-specified probabilistic neural network[21] degenerates to an inefficient neighborhood classifier.[22] This is not unexpected, since both seek to estimate the local probability density function for the multivariate classes in a data set. As another example, the multiple nearest-neighbor rule can be implemented in a network configuration.[23]

### 1.3 *Research Context*

The relationship between neighbor-based classifiers and neural networks led Serpico and Roli[19] to devise an experiment that included a comparison between a probabilistic neural network, a feedforward network, and a multiple nearest-neighbor classifier. Hardin also compared parametric and nonparametric classifiers in a hybrid classification scheme.[1] Building on the work of those authors, this research validates that work and adds to our understanding of neural networks and neighbor-based classifiers in the following ways.

1. Whereas Serpico and Roli studied supervised classification of airborne optical and SAR imagery,[19] the focus of this research is the semisupervised classification of landcover using Landsat TM imagery. Because hand grouping of clusters in the semisupervised process can generate multispectral clusters with nonelliptical or disjoint shapes patently unsuitable for parametric classifiers, neural networks and nonparametric neighbor-based classifiers appear promising and should be investigated.
2. Hardin's experiments neglected the third step in the hybrid classification process described previously—hand regrouping of clusters prior to the pixel assignment phase. Since quick clustering algorithms using mean and covariance relationships frequently yield clusters with ideal shape characteristics for later application of parametric classifiers,[24] Hardin's classification problem using these clusters was not difficult for the parametric classifiers he employed in his research. In contrast, the classification problem studied herein—that of assigning pixels to hand-regrouped cluster classes—is much thornier, particularly for parametric classifiers. Furthermore, Hardin's experiments did not include classifiers from the neural network family.[1]
3. The experiment conducted by Serpico and Roli was conducted on a single scene.[19] The validation provided by conducting similar experiments on several other images contributes to our body of knowledge as practitioners venture to formulate general guidelines for the use of nonparametric and network-based classifiers.
4. The ML classifier is neither neighbor-based nor neural-network-based, yet its performance on the six TM images is also documented in this report. The results thus provide researchers another reference comparing parametric classification with neural network approaches.

### 1.4 *Research Objectives*

The objectives of this research were to determine which of several neighbor-based and network classifiers produced the highest accuracy in the pixel assignment step of semisupervised classification. Formulated on the results of previous research, the following hypotheses were tested:

- *Hypothesis 1:* Given the theoretical relationship between the neighborhood-based classifiers and the probabilistic neural network, their accuracy will be comparable.
- *Hypothesis 2:* The accuracies within the two groups of classifiers will be very similar. In other words, all the neighbor-based classifiers will produce equivalent accuracy. Furthermore, the two network-based systems will produce similar accuracy.
- *Hypothesis 3:* At a minimum, the accuracy of all the classifiers will be as good as that of the parametric ML method.
- *Hypothesis 4:* The accuracy of the feedforward network will not significantly exceed the accuracy of the neighborhood-based classifiers.

These hypotheses were tested by selecting six Landsat Thematic Mapper (TM) images representing a variety of landcover in the United States and conducting a semisupervised classification separately on each. As part of the pixel assignment step of the process, seven classifiers were used and compared for final accuracy of class labeling. The seven classifiers included four neighborhood-based classifiers and two neural network approaches.

The research presented in this paper should be interpreted in the context of its limitations. Six images are not representative of Earth's entire surface, neither do seven algorithms exhaust all the possible classification approaches. The classifiers themselves also have changeable parameters that produce different classification results. In summary, when image pixel assignment accuracy is the goodness criterion, the *universal* superiority of one algorithm over alternatives cannot be claimed for remote-sensing image classification. High classification accuracy primarily depends on the characteristics of the image being analyzed, careful preprocessing manipulation, and due attention to classifier design and parameter choice. (The author is indebted to the anonymous reviewers for their insight into this matter.)

The remainder of this paper has six sections. First, the TM data sets are introduced, followed by an outline of the experimental methodology. After that outline, the neighborhood-based classifiers used in the study are specified, as well as the neural network classifiers. Results and conclusions finish the paper.

## 2 Image Data Sets

Because they are dependent on a single scene, results from comparative studies in remote sensing are seldom generalized. For this reason, six TM images (Fig. 1) acquired during the 1980s representing several different landcover types were selected for repetitive application of a semisupervised classification experiment. The particular scenes were selected for two primary reasons. First, purchased as educational data sets, they are inexpensive and available to other researchers. Second, they contain the requisite diversity in landcover that the experiment required. Obtained from several regions of the USA (Fig. 2), the images provided cover-type examples ranging from coastal wetlands to hot desert. Third, they had been employed in other comparative experiments.[1]

1. *Latour (January 1983):* This is an agricultural scene near Helena, Arkansas, USA, centered on the Mississippi River. The spectral differences in this winter scene are primarily due to differences in soil moisture and the presence of winter crops, weeds, forest, and fluvial features associated with the sparsely vegetated river bottomland.
2. *Morro Bay (November 1984):* This image extends from Morro Bay, California, USA, into the Pacific Coast Range. Natural vegetation includes lower-elevation chapparral, higher-elevation hardwoods, and wetlands along the coast. Some agriculture is present, as well as landcover classes typical of a west-coast suburban area.
3. *San Joaquin (September 1986):* This scene covers a portion of the San Joaquin Valley near Bakersfield, California, USA. It depicts high-value agricultural crops in various stages of growth, as well as highways, orchards, oil fields, and small bodies of water.
4. *Little Colorado (August 1985):* Centered on the junction of the Little Colorado and Colorado rivers, this image shows the region around the Grand Canyon, Arizona, USA. Most of the variation in the image is due to differences in surficial geology. There is little apparent influence of vegetation in this late-summer image.
5. *New Orleans (March 1985):* This image covers northern New Orleans, Louisiana, USA, along its shoreline with Lake Pontchartrain. Most of the image depicts coastline and urban and suburban landcover, as well as agricultural and wetland vegetation.
6. *Black Hills (May 1985):* This image depicts the mining region adjacent to Lead, South Dakota, USA. The image contains obvious evidence of active lead mining and abandoned mining areas in diverse stages of secondary growth or reclamation. Dryland small-grains agriculture also appears, intermixed with patches of rangeland. Forests of ponderosa pine are visible on the higher mountain slopes where mining is absent.

In the experiments that are described below, all seven TM bands of the $512 \times 512$ images were used in the clustering and classification process. Although the normal procedure is to preface classification with some sort of band elimination, feature extraction, or dimension reduction, this usual preliminary step was left undone in order to remove one variable from the experiment.

## 3 Experimental Methodology

In the process of taking satellite imagery and creating a landcover map, several subjective decisions must be made by the practitioner. The choices are based on preference,

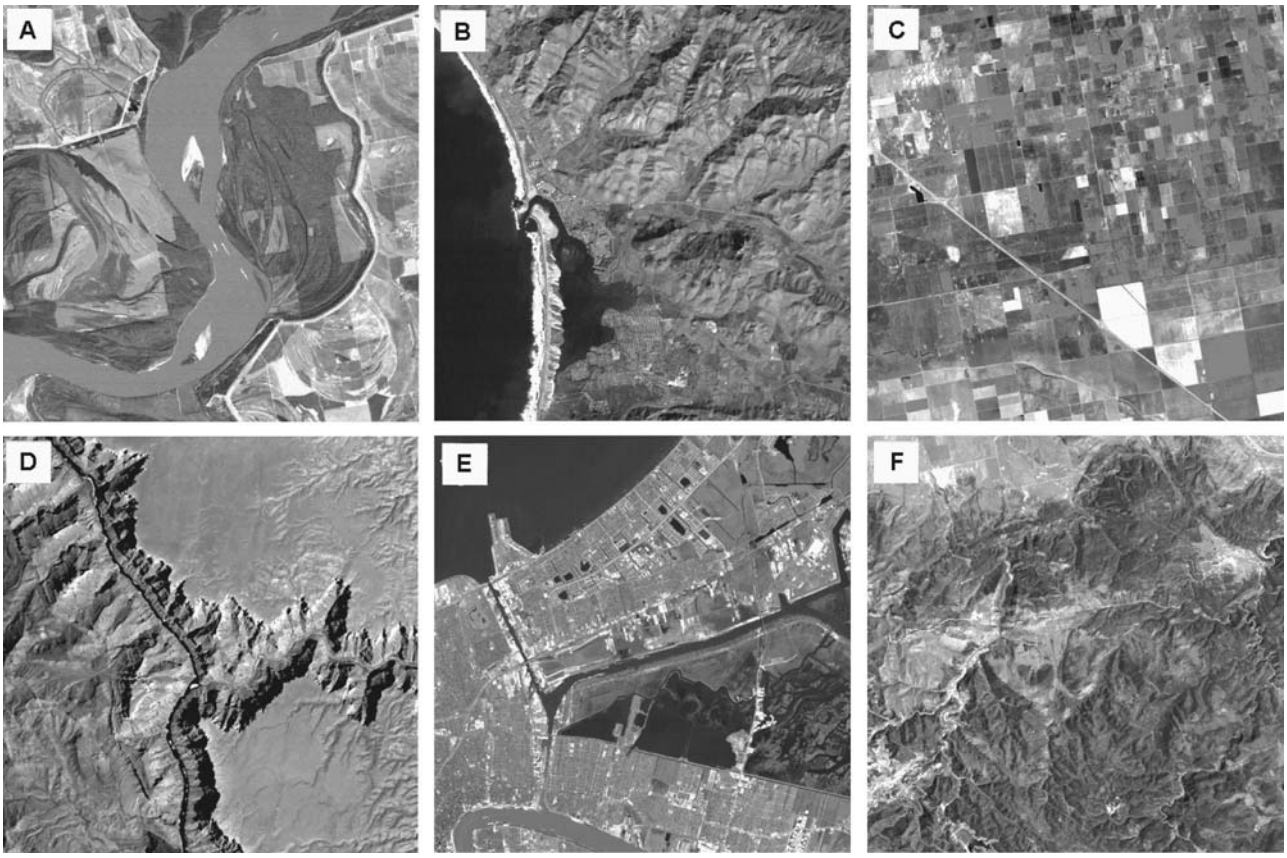


**Fig. 1** Monochrome band composites of the test images: A, Latour, AR; B, Morro Bay, CA; C, San Joaquin, CA; D, Little Colorado, AZ; E, New Orleans, LA; F, Black Hills, SD.

policy, prejudice, or trial and error. Regardless of the selection criteria, each of the decisions affects the character, accuracy, and utility of the final map product. Generally, the variety of options available allows an optimal product to be generated. However, in comparative studies such as this, valid generalization of the experimental results from six different scenes is possible only if the methodology is more rigidly established. In the experiments described below, the goal was to employ a methodology that was repeatable from image to image by any researcher who followed the guidelines. Where an exception to the methodology was raised and required intervention, there was a concerted effort to avoid prejudicing the final results.

The experiment can be reduced to three general procedures. These three procedures were applied in turn to each of the six EOSAT scenes separately. The procedures included (1) sampling and clustering, (2) training and testing, and (3) accuracy assessment.

The method of creating the training data for the various classifiers can be described in the following steps:

1. A sample of 15,000 pixels was randomly extracted from the image. All seven TM bands were retained.
2. The sample was submitted to the SPSS (version 8.0) QuickCluster routine. Depending on initial opinions about the scene content, a cluster solution of between 18 and 36 clusters was selected. Where more classes were apparent after viewing false-color composites of the image, more initial clusters were requested. Recognizing that this was the most subjective phase of the whole experiment, preexperimental trials were conducted to determine how the decision influenced the final landcover groups. In summary, the final classes produced in step 4 were quite invariant to the number of original clusters requested.
3. Clusters with large numbers of pixels were wholly retained. No other groups were added to these large clusters. The pixel count required to define one of

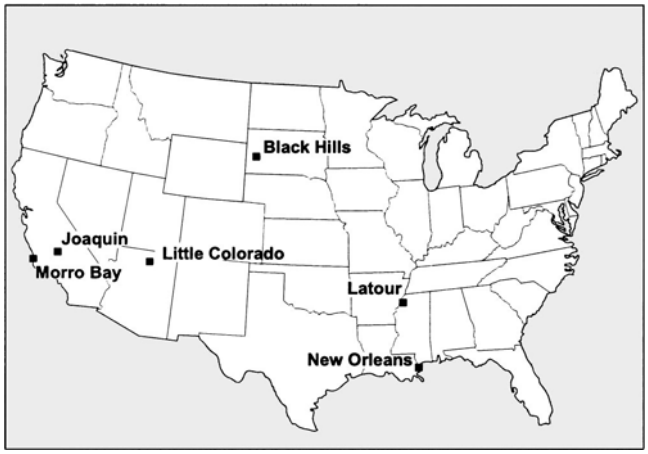


**Fig. 2** The locations of the six test sites.

these groups varied with each cluster solution. If the cluster solution produced a group of 1000+ pixels, the value of 1000 became the cutoff. When solutions produced no group with 1000+ pixels, but rather produced clusters with only 900+, then 900 pixels became the cutoff.

4. After the large clusters were set aside, the following process was used to roughly approximate the hand grouping phase of the semisupervised process. The centroid of each cluster was calculated, and each cluster's closest relative in Euclidean space was identified. Each cluster was then blindly placed in the same group as its closest neighboring cluster. In some cases, only a pair of the original clusters constituted a final group, whereas in other cases this simple rule chained several together into the same group. There were isolated occasions where this chaining produced final groupings with too few classes (or classes with too many pixels) to be an interesting classification problem. In these cases, the degenerate solution was discarded and the cluster regrouping was repeated. However, in the second attempt, the two cluster neighbors with the largest intervening Euclidean distance were forbidden to combine. If this did not mitigate the chaining problem adequately, the next farthest cluster neighbors were likewise prevented from mating. This pattern of intervention was successively applied until a reasonable number of final groups were produced.
5. Now grouped into final classes, the 15,000 pixels were randomly split into training and test sets of 5000 and 10,000 pixels respectively. It is significant that this random assignment combined with random sampling of step 1 ensured training classes with pixel proportions equal to the actual proportions in both the originating TM image and the 10,000-pixel test data set.

After the sampling, clustering, combining, and splitting steps were complete, the seven classifiers described below were trained using the 5000-pixel subset. The 10,000-pixel test data set was then submitted to each trained classifier, producing a confusion matrix, which could be analyzed for classifier performance. Several metrics were used to assess the classifier accuracy, including (1) overall matrix accuracy percentage, (2) average classwise matrix accuracy, and (3) Cohen's overall matrix kappa. Since the story told by each metric was substantially the same, the discussion in this paper will focus on Cohen's overall matrix kappa and its related tests of significance.

## 4 Neighbor-Based Classifiers Studied

There is a large body of literature devoted to neighbor-based classifiers. Several different definitions of neighborhood are possible, and several variations on each general rule are conceivable. In the interest of economy, this presentation will be limited to those neighbor-based classifiers used in the experiment.

### 4.1 *k-Nearest-Neighbor Rule*

Fix and Hodges are credited with the first formulation of nearest-neighbor rules, developing the *k*-nearest neighbor (KNN) rule as an attempt to nonparametrically model multivariate density functions.[25] Applied to remote sensing, the KNN rule states that an unlabeled pixel assumes the identity represented by the majority class of its *k* nearest neighbors. (In the event of a tie, the pixel can be assigned randomly to one of the tied classes.) It is important to remember that the neighborhood is represented in the multivariate spectral coordinate space of the training data. While the logic of the rule is obvious—a pixel of a given class is most likely to reside in the near neighborhood of pixels from the same class—it is not obvious that the KNN rule is also a ML classifier if the proportion of pixels in each class is represented by the same proportion of pixels in the training set.[26,27] As mentioned previously in Sec. 3, this requirement is satisfied by the training data.

The important parameter that the analyst must specify before using the KNN rule as a classifier is the value of *k*—the number of pixels to define a neighborhood. No firm rule exists to determine what value of *k* may produce the highest accuracy for a given classification problem, and experimentation is warranted. Because of this, in this research, *k* was systematically increased on the interval {3, 6, 9,..., 42}. The value of *k* producing the highest accuracy was retained.

### 4.2 *First-Nearest-Neighbor Rule*

The first-nearest-neighbor (FNN) rule is the logical reduction of the KNN method—*k* becomes one. In this project it is considered a separate classifier because of its popularity in the social sciences, ease of use, and good performance.[1] In image processing, the FNN rule forces an unlabeled pixel into the same class as its nearest spectral neighbor in the training set. Although other distance measures are possible, the use of Euclidean space is widespread in image processing because it mitigates the computational burden inherent in alternative complicated formulas.

Like any member of the KNN family, the FNN classifier is a ML classifier when the same constraint regarding training pixel proportions is satisfied.[27] Since the neighborhood is predefined as the sole nearest neighbor, no other parameters require specification, and there is no need to determine a tie-breaking rule.

### 4.3 *Distance-Weighted Neighborhood Rule*

As mentioned above, the KNN rule labels an unclassified pixel by taking a simple vote of the pixel's *k* nearest neighbors. The query pixel is assigned to the class represented by the majority vote. Each of the *k* pixels carries an equal vote, regardless of their distance from the query pixel. As an enhancement to this algorithm, Dudani[28] proposed that the KNN rule take into consideration the actual spectral distance between the query pixel and each of the *k* pixel neighbors. Dudani reasoned that the pixels closest to the query pixel should cast votes with more weight than pixels more distant. Thus, although the neighborhood of this distance-weighted neighborhood (DWN) classifier is the same as for the KNN rule, the decision criterion places an unlabeled pixel in the class with the highest total weighted vote.[28]

In utilizing the DWN approach for image classification, the analyst must specify the weighting function in addition to the number of pixels necessary to define a neighborhood.

Macleod et al.[29] commented that any results obtained from the classifier would therefore depend heavily on the weighting function used. Since this classifier has seen little use in remote sensing, the available literature is silent on the best function to use in image processing. In this research, the vote of each training pixel among the $k$ neighbors was weighted according to the inverse of its squared distance from the query pixel. As this is a heuristic with no foundation in statistical probability theory, any success in the estimation of the local probability density function performed by this weighting rule is serendipitous rather than designed. Recognizing that other alternatives may be better estimators, the inverse-square weighting rule was chosen in this research because it (1) was computationally efficient, (2) is easily understood, and (3) produced very high accuracy in the final tests.

Since real arithmetic was used in calculating the pixel distances in the DWN rule, ties would be infrequent, but another exception would require handling. If a pixel in the training set had the same band values as the query pixel, the weight of its vote would be infinite, swamping the contribution of other neighboring pixels. In these cases the query pixel was randomly assigned to one of classes represented among the $k$ neighbors.

### 4.4 *Bayesian Nearest-Neighbor Rule*

Assume a population of $N$ pixels. Draw a simple random training set of $n$ pixels composed of $g$ classes ($i=1,...,g$), each class having a prior probability $p_i$ of membership in the larger population. Assume also that query pixel $j$($j=1,...,N$) has a multivariate vector of measurements $\mathbf{x}_j$. Bayes's theorem then takes the common form

$$p(h|\mathbf{x}_j)=\frac{p(\mathbf{x}_j|h)p_h}{\Sigma_{i=1}^{g}p(\mathbf{x}_j|i)p_i}, \tag{1}$$

where $p(h|\mathbf{x}_j)$ is the posterior probability that pixel $j$ belongs to class $h$ given its measurement vector $\mathbf{x}_j$.

Consider now a modification of the KNN rule. Recall that the number of neighbors to serve in a KNN classification algorithm is called $k$. In the process of labeling unknown pixel $j$, the number of votes cast for class $h$ can be denoted as $K_h$. As reviewed by James,[27] Bayes's rule can be calculated by

$$p(h|\mathbf{x}_j)=\frac{K_h p_h}{\Sigma_{i=1}^{g}K_i p_i}. \tag{2}$$

This equation summarizes the logic behind the Bayesian nearest-neighbor rule. Applied to a semisupervised classification problem, the $k$ nearest neighbors to the query pixel are first identified, and the number of neighbors in each of the $g$ landcover classes is then tallied. However, unlike the KNN rule, the voting does not end there. In addition, the value of $p(h|\mathbf{x}_j)$ is calculated for each class, and the query pixel is assigned to the class that generates the largest value. As with any other Bayesian classifier, an unlabeled pixel would be assigned to the class that generated the highest *a posteriori* probability estimate. In the experiments described below, the same $k$ neighbors used in the KNN algorithm were also used in the Bayesian classifier.

The relationship between this classifier and the KNN rule is clear. Its primary advantage over its predecessor is its ability to explicitly incorporate prior probabilities into the classification process. However, because it must compute Eq. (2) for every group, slower execution than the KNN algorithm would be expected.

## 5 Network-Based Classifiers Studied

Like parametric classifiers and neighborhood-based classification algorithms, there exists a variety of network-based classification methods with different network paradigms and architectures. Unfortunately, only a few have remained popular since their introduction. This research examined two: the common feedforward network and the probabilistic neural network.

### 5.1 *Feedforward Network*

As observed by Serpico and Roli,[19] feedforward neural networks consisting of multilayer perceptrons are the ''most widely used neural networks for the classification of remotely sensed images.'' The perceptron introduced and examined by Rosenblatt[30] is the parent of this modern feedforward network. Like its modern descendent, the first perceptron had multiple layers, learned by successive presentation of patterns, and used an iterative algorithm that was designed to adjust weights until convergence on the optimum solution was achieved. Its greatest deficiency was its ability to only solve linearly separable classification problems.[31] After Rosenblatt's tragic death in 1971, progress in neural network research was stymied until David Rumelhart[32] and others developed a multilayer feedforward network, utilizing backpropagation of errors, that could classify nonlinear data.

The general character of the feedforward network and its application to remote-sensing classification problems has been extensively reviewed[12] and needs no elaboration here. For an experiment purporting to be an objective comparison of classification methods, issues relating to network architecture and operational parameters do however demand recapitulation. Adopting the excellent work of Miller et al.[13] as the exemplar of complete disclosure, subjects relating to number of layers, layer node counts, data format and scaling, learning rule, and learning schedule will be recited.

It is now common knowledge that only one hidden layer is required to represent any continuous function, as long as sufficient nodes are present.[33] Methods of estimating the necessary node count abound,[10,34,35] as well as approaches to pruning nodes from the network to make it more efficient.[36] Despite these rules, trials with the six images used in these experiments revealed that a network with two hidden layers provided faster training, better generalization, and higher accuracy than a single-hidden-layer model. This could be attributed to the greater degrees of freedom in a two-layer network and the disjoint nature of some of the classes created when the clusters were regrouped. Following the usual convention, one input neuron was reserved for each feature (a single TM band). The optimum hidden-layer node counts were found by brute force. Training and testing were done with all possible hidden-node counts from the set {3, 6, 9, 12,..., 42}. The network configuration that produced the highest kappa on the 10,000-pixel test set

(after 2500 complete passes of the training set through the network) was retained. Both hidden layers were assigned the same node count to simplify matters. The common sigmoid activation function was used on all layers.

When training the network, neuron weights were initially assigned with random numbers. Naive steepest-descent algorithms located minima. While there were initial concerns about local-minimum problems, several random reseeding and retraining repetitions on the images indicated that either the same local minimum was being obtained or a true global minimum value was achieved.

The learning rule for updating a particular neuron weight at iteration $i(\omega_i)$ was

$$\omega_i = \omega_{i-1} + \Delta_i, \qquad \text{where} \quad \Delta_i = \beta \epsilon x + \alpha \Delta_{i-1}. \tag{3}$$

Here $\beta$ and $\alpha$ are learning and momentum parameter, respectively, $x$ is equal to the input value for the neuron, and $\varepsilon$ is the output error of the neuron. Except for the trials involving very few neurons in the hidden layer, the initial value for learning was in the range 0.00125 to 0.0025, and the momentum was maintained between 0.05 and 0.5. Weights were updated after the passage of each training pattern through the network.

Data scaling has a profound impact on feedforward-network training.[31] In these experiments, each TM band was scaled separately by finding is minimum and maximum and then linearly compressing it between zero and one, using double-precision variables. This was predetermined, because the range coincided with the logistic activation function output. No other scaling method was attempted, because final test results indicated it did not need fixing.

Patterns for the output neurons were encoded according to the usual convention of using an individual neuron for each nominal class in the data set.[31] For example, if the image being classified required discrimination between six landcover types, the network was provided with six output neurons. The output training pattern for class 3 would be represented by a value of 0.95 on the third neuron among the six and a value of 0.05 on the remaining five. In the testing phase, an unlabeled pixel was assigned to the class represented by the neuron producing the highest activation.

### 5.2 *Probabilistic Neural Network*

Placing the theoretical work of Meisel[37] in a neural network context, Specht's probabilistic artificial neural network[21] has distinct advantages over the feedforward network. Strongly founded on Bayesian decision theory, perhaps its greatest advantage to classification is its capability to provide confidence levels of group membership as part of the classification process.[22] At the heart of this capability is either Parzen's method for calculating an estimated probability density function[38] or Cacoullos's multivariate extension.[39] For a random sample of size $n$, the point density function for a single variable $f(x)$ with cases in the set $m = x_1, x_2, \ldots, x_n$ can be estimated by

$$f(x) = \frac{1}{n\sigma} \sum_{m=1}^{n} W\left(\frac{x - x_m}{\sigma}\right), \tag{4}$$

**Table 1** Summary of the final class groupings used in the experiments. The lowest average Bhattacharya distance was produced by the Black Hills experiment, indicating that it would be the most difficult classification problem.

| Image | Number of groups ($g$) | Intergroup distance: Bhattacharya simple average | Intergroup distance: Bhattacharya Average weighted by priors |
|---|---|---|---|
| Latour | 9 | 14.98 | 7.83 |
| Morro Bay | 11 | 20.84 | 10.82 |
| San Joaquin | 12 | 16.30 | 5.30 |
| Little Colorado | 11 | 18.71 | 5.61 |
| New Orleans | 8 | 11.13 | 4.86 |
| Black Hills | 6 | 5.42 | 1.42 |

where $W$ represents a weighting function (potential function or kernel) to describe the overall shape of the distribution centered on a training value, and $\sigma$ represents the width of the distribution. Since most probabilistic neural networks utilize a Gaussian function (or a close equivalent) for convenience,[22] the most important design decision affecting the classification accuracy of a probabilistic neural net is the value of $\sigma$. If $\sigma$ is too small, the network becomes an inefficient neighborhood classifier with no generalization capability.[22] If $\sigma$ is too large, details in the density function are lost through overgeneralization.

For each of the images in our experiments, 30 random values between 0.003 and 5.0 were tried in order to locate the broad interval containing the optimum $\sigma$. Successive reduction of this interval was performed until the classification accuracy no longer improved. This iterative process is the equivalent of training for the feedforward network. To simplify matters, each variable and class used the same value of $\sigma$. Prior probability information was not explicitly incorporated in the architecture of the probabilistic neural network classifier.

As in the feedforward network, the TM image data were scaled before submitting them to the probabilistic neural network. Adopting recommendations of Masters,[22] the mean and standard deviation were calculated individually for each TM band. Each image value in each band was then converted to its $z$-score equivalent with the calculated statistics.

## 6 Results and Discussion

Table 1 summarizes the results from the cluster grouping. Final group counts varied from 6 for the Black Hills image to 12 for the San Joaquin image. The third and fourth columns are the intergroup distances for the final landcover classes. The intergroup measures used for the two columns are the average unweighted Bhattacharya distance and the average Bhattacharya distance weighted by prior probabilities, respectively.[40] In either case, higher values indicate more average separation between the group centroids (accounting for group dispersion as well). Judging solely by the intergroup distance criteria, it appears that Black Hills

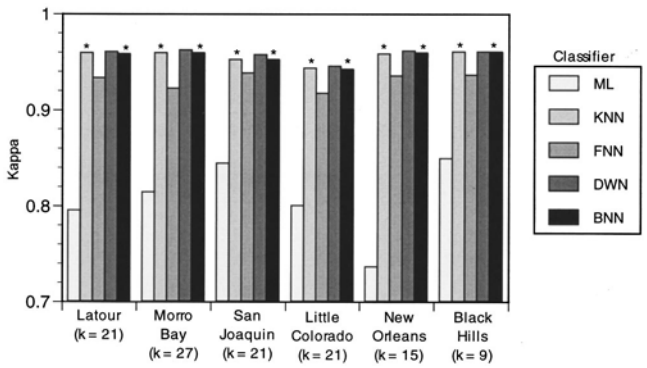


**Fig. 3** Kappa values for the neighborhood-based classifier experiments. Kappa values statistically equal to the highest ($\alpha = 0.05$) are shown by an asterisk. Here $k$ is the number of neighbors used in the neighbor-based classifiers.

presented the most difficult discrimination problem, whereas the Morro Bay image generated the easiest problem.

Figures 3 and 4 display the results of the 42 (six images×seven classifiers) classification experiments. The classification producing the highest raw kappa value for each image is obvious, and kappa values statistically equivalent to it ($\alpha = 0.05$) are shown by an asterisk. Surprisingly, despite the great difference in interclass distances, the discrimination of all the classifiers was generally adequate to excellent. Although discrimination was anticipated to be superior for Morro Bay and worse for the Black Hills image, the final accuracy of the two was almost identical for several of the classifiers tested.

### 6.1 *The Neighborhood-Based Classifiers*

In the preexperimental trials, no significant improvement in accuracy was achieved using various values of $k$ from the set $\{3, 6, 9, \ldots, 42\}$ once a lower limit of $k$ was reached. However, as shown along the $x$-axis of Fig. 3, the actual optimal value of $k$ differed from image to image. That value ranged from 27 for the Morro Bay image to 9 for the Black Hills image. Regarding the best choice of $k$, the only guidelines gleaned from these tests were (1) to choose a value much smaller than the lowest count of pixels among the several training classes and (2) to choose an odd value to minimize ties. The trial results revealed that ambiguities arising from ties occurred infrequently (<1%) when $k$ was an odd number.

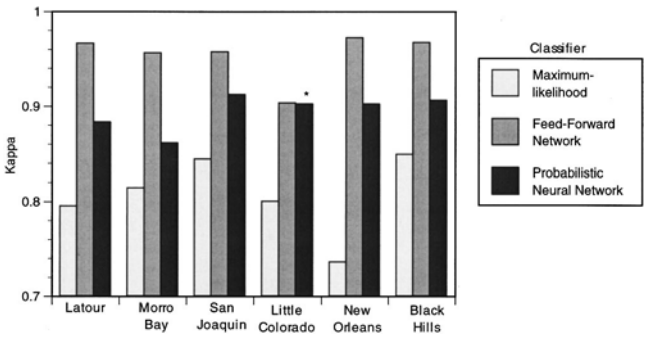


**Fig. 4** Kappa values for the network-based classifier experiments. Kappa values statistically equal to the highest ($\alpha = 0.05$) are shown by an asterisk.

As measured by Cohen's kappa, all the neighborhood-based classifiers exceeded 0.91 accuracy on all the test images (see Fig. 3). The highest classification accuracy (0.962) was achieved with the DWN classifier on the Morro Bay image, while the FNN classifier applied to the Colorado image produced the lowest accuracy (0.917). When the images are considered separately, there were several statistically significant differences between the neighborhood-based classifiers as measured by their tablewise kappa values, but the practical differences were minor. Because the group proportions in the training data were equal to the same group proportions in the test data, the use of prior probabilities did not advantage the BNN classifier in relation to the KNN rule, and the kappas produced in each image were never significantly different.

In all of the images, the DWN classifier produced the highest raw kappa. There was also no statistically significant difference between the accuracy produced by the DWN classifier and the accuracy of the KNN rule. The same was true of the BNN classifier. In every case it produced results statistically equivalent to the most accurate classifier. Furthermore, without exception the FNN classifier produced lowest kappa values among the classifier set—the FNN classifier was never statistically equivalent to the highest accuracy achieved by the DWN, BNN, or KNN classifier.

It is also apparent that the accuracy produced by the neighborhood-based algorithms was always superior to the ML classifier. In several cases, the accuracy difference between this parametric approach and the nonparametric methods exceeded 10 percentage points, and in no case was it less than 5 percentage points. In all cases the difference between the neighborhood-based classifiers and the ML classifier were statistically significant ($\alpha = 0.05$).

### 6.2 *The Network-Based Classifiers*

As presented above, two artificial neural-network-based classifiers were compared. The first was the usual feedforward backpropagation network, whereas the second was the probabilistic neural network. Like the nonparametric classifiers tested, both neural network types produced very high classification rates (see Fig. 4). Except for the Little Colorado image, the backpropagation classification was superior to the probabilistic neural network in every case. These differences were all significant at an $\alpha$ level of 0.05. The greatest difference between the two was in classifying the Morro Bay imagery, where the probabilistic neural net produced a kappa of only 0.861, compared to 0.956 for the backpropagation network.

As expected, the number of hidden neurons required to obtain an optimum classification differed according to the image (see Table 2). Table 2 also shows that the accuracy improvement between successive hidden-node solutions was insignificant after a certain node count in each hidden layer was reached. Speaking roughly, this cutoff was 1.5 to 3 times the number of landcover classes needing discrimination. The Little Colorado image required the greatest number of neurons (36) for successful classification, whereas three images required only 18 neurons in each hid-

**Table 2** Rounded tablewise kappa values obtained with different hidden-layer neuron counts. The solution retained in shown by an asterisk. In the headings, $g$ is the number of groups in the classification problem.

| | Kappa | | | | | |
|---|---|---|---|---|---|---|
| Neurons in each hidden layer | Latour ($g=9$) | Morro Bay ($g=11$) | San Joaquin ($g=12$) | Little Colorado ($g=11$) | New Orleans ($g=8$) | Black Hills ($g=6$) |
| 3 | 0.41 | 0.68 | 0.36 | 0.54 | 0.21 | 0.61 |
| 6 | 0.84 | 0.76 | 0.72 | 0.76 | 0.70 | 0.94 |
| 9 | 0.95 | 0.87 | 0.90 | 0.75 | 0.93 | 0.95 |
| 12 | 0.95 | 0.93 | 0.93 | 0.74 | 0.96 | 0.95 |
| 15 | 0.94 | 0.93 | 0.94 | 0.77 | 0.95 | 0.96 |
| 18 | *0.97 | 0.94 | *0.96 | 0.80 | *0.97 | 0.96 |
| 21 | 0.95 | 0.95 | 0.95 | 0.84 | 0.97 | *0.97 |
| 24 | 0.96 | *0.96 | 0.95 | 0.85 | 0.97 | 0.96 |
| 27 | 0.96 | 0.95 | 0.95 | 0.87 | 0.97 | 0.97 |
| 30 | 0.96 | 0.96 | 0.96 | 0.88 | 0.97 | 0.97 |
| 33 | 0.96 | 0.96 | 0.96 | 0.84 | 0.97 | 0.97 |
| 36 | 0.96 | 0.96 | 0.96 | *0.90 | 0.97 | 0.97 |
| 39 | 0.96 | 0.96 | 0.96 | 0.89 | 0.98 | 0.97 |
| 42 | 0.97 | 0.96 | 0.96 | 0.89 | 0.97 | 0.97 |

den layer for comparable accuracy. The reason for the large difference remains unclear, especially when it is remembered that all these images had approximately the same number of classes.

## 7 Conclusion

In Sec. 1.4, four hypotheses were presented. The first hypothesis stated that the accuracy of the probabilistic neural network would be equivalent to the accuracy of the neighborhood-based classifiers. The experiments proved this hypothesis false. In fact, in every case, the worst neighbor-based classifier produced significantly higher accuracy than the probabilistic neural neighbor. Apparently, although both approaches estimated the multivariate density function of the semisupervised training set, the neighborhood-based methods estimated it more accurately.

The second hypothesis implied that all the neighborhood-based classifiers should produce equivalent accuracy values. Likewise, the probabilistic and feedforward networks should have produced equal kappa values. The experiments demonstrated that the KNN, DWN, and BNN rules produced nearly identical accuracy—there was never a statistically significant difference ($\alpha=0.05$) between them. However, the accuracy of the FNN classifier was always significantly lower. The second hypothesis was more clearly wrong in postulating that the two network-based methods would be equivalent as well—with one exception, the feedforward network always produced higher accuracy than the probabilistic neural network.

The third hypothesis stated that all the classifiers would produce accuracy at least equivalent to the ML classifier. The conclusion can be stated in stronger terms. All the classifiers tested produced tablewise accuracy significantly better (both in statistical and substantive terms) than the ML classifier.

The fourth hypothesis had the greatest applied importance because it compared the popular feedforward network with the neighborhood-based classifier family. The statement read ''the accuracy of the feedforward network will not significantly exceed the accuracy of the neighborhood-based classifiers.'' The resolution of this hypothesis depends on whether significance is measured in statistical or practical terms, and the results are not conclusive. The best neighbor-based classifier (DWN) is compared with the feedforward network in Table 3. Statistically speaking, on three of the images the feedforward network produced higher pixel assignment accuracy than the neighborhood-based classifiers, but on two images the opposite was true (Table 3). The accuracies produced by the two methods on the San Joaquin image were statistically equivalent. From a practical perspective, the accuracies achieved by the two methods are nearly identical—never did the tablewise accuracies of the two methods differ by more than a few percent.

### 7.1 *Operational Guidelines*

From these results, a few simple operational guidelines for classifier use can be suggested. First, practitioners should try a feedforward neural network classifier to see if the produced accuracy is higher than the parametric ML alternative. In the exploratory phase, the initial network configuration might include two layers with a dozen hidden neurons each. If the accuracy of the feedforward network is encouraging, the hidden-neuron count can be doubled to approach the highest possible accuracy.

While the feedforward network shows great promise for semisupervised classification, experimentation with neighborhood-based classifiers is also warranted, particularly in that they require much less experimentation than feedforward networks to obtain nearly equivalent results.

**Table 3** Comparison of distance-weighted neighbor rule with the results of the feedforward neural network. For each image, the winning kappa value is emboldened. Except for the San Joaquin image, the differences are statistically significant ($\alpha = 0.05$).

| | Kappa | | | | | |
|---|---|---|---|---|---|---|
| Classifier | Latour | Morro Bay | San Joaquin | Little Colorado | New Orleans | Black Hills |
| DWN | 0.960 | **0.962** | 0.957 | **0.945** | 0.961 | 0.960 |
| Feedforward network | **0.966** | 0.956 | **0.957** | 0.903 | **0.972** | **0.967** |
| *Z*-test | 2.09 | −2.01 | 0.07 | −10.42 | 3.00 | 2.43 |
| Significance (two-tailed) | 0.037 | 0.044 | 0.944 | <0.001 | 0.003 | 0.015 |

Whereas the action of the feedforward network is governed by several parameters needing to be tweaked to achieve optimum results, the neighborhood-based classifiers require only a few. These included such simple parameters as the number of neighbors ($k$) to use, the group prior probabilities, and a rule to handle voting ties.

Of the three parameters listed above, choosing the correct value of $k$ is critical. As mentioned previously, choosing an odd value minimizes ties. However, the value of $k$ required for the highest pixel assignment accuracy will change with the size of the training set and perhaps the number of groups to be discriminated. It will probably be best determined by experimentation. Since neighborhood-based classifiers do not have a computationally intensive training phase, it is simple to exhaustively try several values of $k$ within the set {1, 3, 6, 12, 24, 48} until high classification accuracy is obtained. From that starting point, the value of $k$ can then be successively refined. If the experimentation indicates that several values of $k$ produce equivalently high accuracy, the smallest possible value of $k$ is preferred. By virtue of the majority-vote rule, large values of $k$ bias assignment to the groups with larger numbers of training pixels. Adopting the smallest possible value of $k$ minimizes this problem.

## References


1. P. J. Hardin, ''Parametric and nearest-neighbor methods for hybrid classification: a comparison of pixel assignment accuracy,'' *Photogramm. Eng. Remote Sens.* **60**(12), 1439–1448 (1994).
2. J. H. Friedman, J. L. Bentley, and R. A. Finkel, ''An algorithm for finding best matches in logarithmic expected time,'' *ACM Trans. Math. Softw.* **3**, 209–226 (1977).
3. M. E. Hodgson, ''Reducing the computational requirements of the minimum-distance classifier,'' *Remote Sens. Environ.* **25**, 117–128 (1988).
4. P. J. Hardin and C. N. Thomson, ''Fast nearest neighbor classification methods for multispectral imagery,'' *Professional Geographer* **44**(2), 191–201 (1992).
5. A. K. Skidmore, ''An expert system classifies eucalypt forest types using Thematic Mapper data and a digital terrain model,'' *Photogramm. Eng. Remote Sens.* **55**(10), 1449–1464 (1989).
6. J. A. Benediktsson, J. R. Sveinsson, and O. K. Ersoy, ''Parallel consensual neural networks,'' *IEEE Trans. Neural Netw.* **8**, 54–64 (1997).
7. G. M. Foody, M. B. McCulloch, and W. B. Yates, ''Classification of remotely sensed data by an artificial neural network: issues related to training data characteristics,'' *Photogramm. Eng. Remote Sens.* **61**, 391–402 (1995).
8. G. F. Hepner, T. Logan, N. Ritter, and N. Bryant, ''Artificial neural network classification using a minimal training set: comparison to conventional supervised classification,'' *Photogramm. Eng. Remote Sens.* **56**(4), 469–473 (1990).
9. J. D. Paola and R. A. Schowengerdt, ''A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery,'' *Int. J. Remote Sens.* **16**(16), 3033–3058 (1995).
10. G. W. H. German and M. N. Gahegan, ''Neural network architectures for the classification of temporal image sequences,'' *Comput. Geosci.* **22**(9), 969–979 (1996).
11. S. R. Chettri, R. F. Cromp, and M. Birmingham, ''Design of neural networks for classification of remotely sensed data,'' *Proc. NASA 1992 Goddard Conf. on Space Applications of Artificial Intell.*, 137–150, Greenbelt, MD (1992).
12. P. M. Atkinson and A. R. L. Tatnall, ''Neural networks in remote sensing,'' *Int. J. Remote Sens.* **18**(4), 699–709 (1997).
13. D. M. Miller, E. J. Kaminsky, and S. Rana, ''Neural network classification of remote-sensing data,'' *Comput. Geosci.* **21**(3), 377–386 (1995).
14. X. Zhuang, B. A. Engel, D. F. Lozano-Garcia, R. N. Fernandez, and C. J. Johannsen, ''Optimization of training data required for neuroclassification,'' *Int. J. Remote Sens.* **15**(16), 3271–3277 (1994).
15. J. D. Paola and R. A. Schowengerdt, ''The effect of neural-network structure on a multispectral land-use/land-cover classification,'' *Photogramm. Eng. Remote Sens.* **63**(5), 535–544 (1997).
16. M. T. Manry, M. S. Dawson, A. K. Fung, S. J. Apollo, L. S. Allen, W. D. Lyle, and W. Gong, ''Fast training of neural networks for remote sensing,'' *Remote Sens. Rev.* **9**(1–2), 77–96 (1994).
17. K. S. Chen, Y. C. Tzeng, C. F. Chen, and W. L. Kao, ''Land-cover classification of multispectral imagery using a dynamic learning neural network,'' *Photogramm. Eng. Remote Sens.* **61**(4), 403–408 (1994).
18. S. D. Murnion, ''Comparison of back propagation and binary diamond neural networks in the classification of a Landsat TM image,'' *Comput. Geosci.* **22**(9), 995–1001 (1996).
19. S. B. Serpico and F. Roli, ''Classification of multisensor remote-sensing images by structured neural networks,'' *IEEE Trans. Geosci. Remote Sens.* **33**, 562–578 (1995).
20. M. E. Voudouri, L. Kurz, and J. M. Kowalski, ''A neural-network approach to nonparametric and robust classification procedures,'' *IEEE Trans. Neural Netw.* **8**, 288–298 (1997).
21. D. Specht, ''Probabilistic neural networks,'' *Neural Networks* **3**, 109–118 (1990).
22. T. Masters, *Advanced Algorithms for Neural Networks: A C++ Sourcebook*, Wiley, New York (1995).
23. Y. Q. Chen, R. I. Damper, and M. S. Nixon, ''On neural-network implementations of *k*-nearest neighbor pattern classifiers,'' *IEEE Trans. Circuits Syst., I: Fundam. Theory Appl.* **44**, 622–629 (1997).
24. H. Spath, *Cluster Analysis Algorithms for Data Reduction and Classification of Objects*, Ellis Horwood, Chichester, England (1980).
25. E. Fix and J. L. Hodges, Project 21-49-004, Report No. 4, USAF School of Aviation Medicine, Randolph Field, Texas (1951).
26. B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, IEEE Computer Society Press, Los Alamitos, CA (1990).
27. M. James, *Classification Algorithms*, Wiley, New York (1985).
28. S. A. Dudani, ''The distance-weighted *k*-nearest neighbour rule,'' *IEEE Trans. Syst. Man Cybern.* **6**(4), 325–327 (1976).
29. J. E. S. Macleod, A. Luk, and D. M. Titterington, ''A re-examination of the distance weighted *k*-nearest neighbor classification rule,'' *IEEE Trans. Syst. Man Cybern.* **17**(4), 689–696 (1987).
30. F. Rosenblatt, ''The perceptron: a probabilistic model for information storage and organization in the brain,'' *Psychol. Rev.* **65**, 386–408 (1958).
31. T. Masters, *Practical Neural Network Recipes in C++, Academic, New York (1994).*
32. J. McClelland and D. Rumelhart, *Explorations in Parallel Distributed Processing*, MIT Press, Cambridge, MA (1988).

33. J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Reading, MA (1991).
34. E. B. Baum and D. Haussler, ''What size network gives valid generalization?'' *Neural Comput.* **1**, 151–160 (1989).
35. D. Fletcher and E. Gross, ''Forecasting with neural networks: an application using bankruptcy data,'' *Inf. Manage.* **24**, 159–167 (1993).
36. Z. Jiang and D. L. Civco, ''Using genetic learning neural networks for spatial decision making in GIS,'' *Photogramm. Eng. Remote Sens.* **62**(11), 1249–1260 (1996).
37. W. Meisel, *Computer-Oriented Approaches to Pattern Recognition*, Academic, New York (1972).
38. E. Parzen, ''On estimation of a probability density function and mode,'' *Ann. Math. Stat.* **33**, 1065–1076 (1962).
39. T. Cacoullos, ''Estimation of a multivariate density,'' *Ann. Inst. Statist. Math. (Tokyo)* **18**(2), 179–189 (1966).
40. R. M. Welch, S. K. Sengupta, A. K. Goroch, P. Rabindra, N. Rangaraj, and M. S. Navar, ''Polar cloud and surface classification using AVHRR imagery: an intercomparison of methods,'' *J. Appl. Meteorol.* **31**, 405–420 (1992).

**Perry J. Hardin** received his PhD from the University of Utah in 1989. He is currently an associate professor of geography at Brigham Young University, where he teaches and serves as the associate director for the Center for Remote Sensing. His research interests include such diverse subjects as land and ice applications of scatterometery, image-processing methods, accuracy assessment, global climate change, tropical deforestation, sustainable tropical agriculture, and the modeling of prehistoric agricultural systems in Central America. This research has resulted in more than a dozen funded research projects and several peer-reviewed publications appearing in journals of engineering, remote sensing, geography, agriculture, and archaeology.