

Land Cover Mapping Using Combination and Ensemble Classifiers

Brian M. Steele¹ and David A. Patterson²

¹Dept. of Mathematical Sciences, University of Montana, Missoula MT 59812;
bsteele@selway.umt.edu

²Dept. of Mathematical Sciences, University of Montana, Missoula MT 59812;
davep@selway.umt.edu

Abstract

In recent years, large scale land cover maps constructed from remotely sensed data have become important information sources for resource management. Classifiers are commonly used to predict land cover for unsampled map units; hence, they play a key role in map construction. Achieving adequate classifier accuracy is often problematic for highly variable and difficult-to-sample landscapes. This article investigates a variety of methods for improving accuracy based on 1) combining a few different classifiers, and 2) creating ensembles of many classifiers. In addition, we derive an analytic expression for the exact bagging k -nearest neighbor classifier.

1 Introduction

A land cover map consists of a set of contiguous map units each of which is labeled according to a land cover class. Land cover classes identify the dominant vegetation (e.g., Douglas-fir forest) or land use (e.g., urban). These maps are used extensively in large-scale resource management and environmental analysis, and play an important role within geographic information systems. A common approach to constructing large-area land cover maps uses remotely sensed data, usually satellite imagery, and a training sample collected by ground visitation. In the case of Landsat Thematic Mapper (TM) satellite imagery, the remotely sensed data consist of measurements of reflectance intensity for 8 bands of the electromagnetic spectrum. Our focus is on constructing classifiers for land cover mapping based on Landsat TM images. Each of the Landsat images covers approximately 34000 km² with a 30 × 30 m pixel lattice, though this pixel size is impractically small for ground sampling and many applications. Consequently, image segmentation is used to replace the pixels by larger polygons. Image segmentation forms polygons by merging adjacent pixels so that within-polygon variation is minimized given certain constraints on polygon size (see Ford et al. 1997 and Ma et al. 2001). The final segmented image is a map consisting of many fewer map units. For our examples, the average number of polygons per scene is about 600,000.

This satellite-based approach to large-scale land cover mapping has the potential to be extremely efficient. However, map utility is diminished by classifier errors induced by factors such as locational and ground-sampling errors and atmospheric distortion. Moreover, most land cover types are constructed by partitioning a continuum of plant communities rather than identifying truly distinct entities. Given these inherent problems, it is critical that the classifiers are as accurate as possible. This article investigates two general approaches to improving classifier accuracy. In the first, a few relatively dissimilar classifiers are combined. Examples of this approach include stacked regression methods (Breiman 1996; LeBlanc and Tibshirani 1996), and the methods of Mojirsheibani (1999) and Steele (2000). Ensemble methods are the second approach; most notably, these methods include bootstrap aggregation (or bagging) (Breiman 1996) and boosting (Friedman et al. 2000; Freund and Schapire 1996; Schapire 1990). The idea behind ensemble methods is to resample the training set and create a new version of the classifier rule. This is repeated many times, and newly produced rules are combined to yield a single rule. There has been extensive research on ensemble methods (see Optiz and Maclin 1999); in particular, there are many variants of boosting. We confine ourselves to the Ada-BoostM.1, or Discrete Ada-Boost algorithm (Freund and Schapire 1997; Friedman et al. 2000).

The principal contributions of this article are a comparison of these methods for four land cover mapping exercises, and the derivation of the exact bagging version of the k -nearest neighbor (k -NN) classifier. The article is organized as follows. Sections 2 and 3 introduce terminology and notation, and describe the conventional, or base classifiers used in this study. Section 4 reviews combination and ensemble methods. Section 5 discusses the Landsat scenes and the training samples, and Section 6 presents the results. The article concludes with Section 7.

2 Notation and Terminology

Suppose that a training sample $X_n = \{x_1, \dots, x_n\}$ of n observations has been collected by sampling a population \mathcal{P} . Each element $x_0 \in \mathcal{P}$ belongs to one of c classes, or groups, identified by the labels $1, \dots, c$. The number of training observations in class g is denoted by n_g . In this study, \mathcal{P} represents a set of map polygons, and $x_0 \in \mathcal{P}$ is a triple $x_0 = (t_0, y_0, z_0)$, where z_0 is a pair of coordinates identifying the location of the polygon centroid, y_0 is the land cover class at z_0 , and t_0 is a multidimensional covariate vector consisting of observations on remotely sensed variables. For all $x_0 \in \mathcal{P}$, t_0 and z_0 are known, whereas y_0 is unknown except for those observations in X_n . The posterior probability that t_0 belongs to class g , given t_0 , is denoted by $P_g(x_0) = P(y_0 = g | t_0)$. A classifier can be viewed as an estimator of $P_1(x_0), \dots, P_c(x_0)$ which assigns x_0 to the class with the largest estimated posterior probability among the estimates $\hat{P}_1(x_0), \dots, \hat{P}_c(x_0)$. That is, if the classifier rule is denoted by η , then $\eta(x_0) = \arg \max_g \hat{P}_g(x_0)$ is the class to which x_0 is classified.

3 Classifiers

Three conventional classifiers are discussed in this article: the k -NN Euclidean distance classifier, a binary tree classifier, and the mean inverse distance (MID) classifier. The estimator of $P_g(x_0)$ produced by a k -NN Euclidean distance classifier is the sample

proportion of the k -nearest neighbors belonging to class g , where the distances between x_0 and the observations in X_n are the Euclidean distances from t_0 to the covariate vectors t_1, \dots, t_n . More formally, let $t_{0,j}$ denote the j th closest observation to t_0 among $\{t_1, \dots, t_n\}$, where the distance between covariate vectors is Euclidean distance, and let $y_{0,j}$ denote the group label of $t_{0,j}$. The k -NN estimate of $P_g(x_0)$ is

$$P_g^{kNN}(x_0) = \frac{1}{k} \sum_{j=1}^k \Psi(y_{0,j} = g), \quad (1)$$

where $\Psi(E)$ is the indicator function of the event E . Ties among the maximum posterior probability estimates may be broken by randomly choosing among the tied groups or by increasing the neighborhood size and recomputing formula (1).

The second conventional classifier is a binary tree classifier (Breiman et al. 1984; Ripley 1996, Ch. 6). A binary tree classifier recursively partitions the covariate space according to a set of binary decision rules. Each split produces two daughter nodes (or subspaces), and those nodes that are not themselves split are called terminal nodes. The rules (or equivalently, the nodes) are determined by examining every possible split on each covariate, and choosing the split that produces the greatest improvement in node purity. Within each terminal node, the estimate of $P_g(x_0)$ is the sample proportion of training observations at the node which belong to class g . In the examples below, trees were formed by holding out a random subset (approximately 10% of the original data set), and constructing the tree using the remaining observations. A terminal node was eligible for splitting only if the node contained at least 20 training observations, and both daughter nodes contained at least 6 observations. Trees were pruned with the objective of minimizing the classification error rate of the tree when applied to the left-out subset.

The third classifier was a spatial classifier developed explicitly for polygon-based land cover mapping. The motivation and methods of spatial classification are discussed by Carpenter et al. (1999), Steele (2000), and Steele and Redmond (2001). Briefly, spatial information is present and potentially useful for classification when land cover class abundance varies with spatially predictable factors such as climate and topography. When the map is a lattice of pixels, patterns of positive spatial association among adjacent map units can be exploited by contextual allocation methods (see Kartikeyan et al. 1994; Lee 2000; Sharma and Sarkar 1998; Stuckens et al. 2000; and Van Deusen, 1995). However, these methods are not appropriate when the map consists of polygons formed by image segmentation because this process tends to produce polygon boundaries that coincide with changes in the values of the remotely sensed variables and land cover class. Consequently, adjacent polygons are not predictably similar with respect to land cover. However, spatial information may be present at a scale larger than adjacent polygons, particularly when the map area is large. For example, the areas covered by the Landsat scenes in this study are large enough to manifest differences in temperature and precipitation related to regional and continental-scale gradients. This variation induces large-scale spatial patterns in the relative frequency of occurrence of land cover classes.

Steele (2000) and Steele and Redmond (2001) have developed an approach to extracting spatial information from the relative abundance and proximity of training observations in the vicinity of an unclassified polygon. Their approach can be motivated by the application of Bayes rule when the conditional probability density functions $p(t_0 | y_0 = g)$ and the prior probabilities $\pi_g = P(y_0 = g)$ are known. Bayes rule minimizes the total probability of misclassification by assigning x_0 to the class with the largest posterior probability

$$P(y_0 = g | t_0) = \frac{\pi_g p(t_0 | y_0 = g)}{\sum_{j=1}^c \pi_j p(t_0 | y_0 = j)}, \quad g = 1, \dots, c,$$

(McLachlan 1992, Ch. 1). The k -NN classifier (equation [1]) is a plug-in version of Bayes rule which estimates $p(t_0 | y_0 = g)$ by the sample proportion of near neighbors belonging to class g , and assumes that $\pi_j = c^{-1}$ for $j = 1, \dots, c$. The assumption of uniform, or non-informative prior probabilities is appropriate if there is no information regarding the class membership probabilities besides that carried by the covariate vectors. In some instances, the relative frequency of occurrence of the c classes in a spatial neighborhood about the location z_0 of x_0 is informative for classification; if so, then this information may be expressed as local prior class probabilities $\pi_g(z_0)$, $g = 1, \dots, c$. We propose a local Bayes rule which assigns x_0 to the class with the largest posterior probability

$$P(y_0 = g | t_0, z_0) = \frac{\pi_g(z_0) p(t_0 | y_0 = g)}{\sum_{j=1}^c \pi_j(z_0) p(t_0 | y_0 = j)}, \quad g = 1, \dots, c. \quad (2)$$

In practice, $\pi_g(z_0)$ and $p(t_0 | y_0 = g)$ are rarely known, so we propose to plug in estimates of $\pi_j(z_0)$ and $p(t_0 | y_0 = j)$, $j = 1, \dots, c$, into formula (2). Estimates of the local priors $\pi_j(z_0)$ are obtained from the mean inverse distance (MID) estimator $\hat{\pi}_g(z_0) = \bar{d}_g(z_0) / \sum_{j=1}^c \bar{d}_j(z_0)$, where

$$\bar{d}_g(z_0) = \frac{1}{n_g} \sum_{i=1}^n \Psi(y_i = g) d^E(z_0, z_i)^{-2}, \quad (3)$$

is the mean inverse squared distance from an observation with location z_0 to group g , and $d^E(z_0, z_i)$ is the Euclidean distance between locations z_0 and z_i . The MID estimates $\hat{\pi}_g(z_0)$, $g = 1, \dots, c$, may be combined with any set of posterior probability estimates. For example, the k -NN+MID classifier estimator of $P(y_0 = g | t_0, z_0)$ is a plug-in version of Bayes rule given by

$$P_g^{k\text{NN}+\text{MID}}(x_0) = \frac{\hat{\pi}_g(z_0) P_g^{k\text{NN}}(t_0)}{\sum_{j=1}^c \hat{\pi}_j(z_0) P_j^{k\text{NN}}(t_0)},$$

and the k -NN+MID classifier rule is

$$\eta^{k\text{NN}+\text{MID}}(x_0) = \arg \max_g P_g^{k\text{NN}+\text{MID}}(x_0).$$

Several remarks are in order. The mean inverse distance $\bar{d}_g(z_0)$ between z_0 and class g is the same as the average illumination generated by the lights of class g at z_0 if the training observations are regarded as lights of equal intensity. The use of the mean inverse distance as a measure of spatial density is not new; for example, see Watson and Philip (1985).

4. Methods of Combining Classifiers

We consider two approaches to combining classification rules. The first approach usually is used to combine a few different classifiers, each of which is constructed from the original data set. We refer to these methods as combination methods, and note that the k -

NN+MID classifier is the first example of a combination classifier. The second approach generates many versions of the same base classifier by resampling the training set, and combines these versions as a single classifier. We adopt a term used within the machine learning community, and refer to these classifiers as ensembles.

4.1 Stacked Regression

The stacked regression method (Breiman 1996; LeBlanc and Tibshirani 1996; Wolpert 1992) combines probability estimates from several classifiers as linear combinations. The method can be summarized as follows. First, the r classifiers, indexed by l , are used to compute class probability estimates $P_g^l(x_i)$, for the $i = 1, \dots, n$ observations and the $g = 1, \dots, c$ classes. Specifically, $P_g^l(x_i)$, $g = 1, \dots, c$, is computed after removing x_i from X_n , using classifier l . These estimates are combined as a row vector

$$P(x_i) = [P_1^1(x_i) \cdots P_c^1(x_i) \cdots P_1^r(x_i) \cdots P_c^r(x_i)], \quad (4)$$

and the calculation is repeated for $i = 1, \dots, n$. Then, all n row vectors are stacked in a $n \times cr$ matrix \mathbf{P} . Let u_g denote an n -vector identifying training set memberships in class g ; i.e., the i th element is

$$u_{ig} = \begin{cases} 1, & \text{if } y_i = g, \\ 0, & \text{if } y_i \neq g. \end{cases} \quad (5)$$

The cr -vector b_g is computed by minimizing the sum of the square errors $S(\beta_g) = (u_g - \mathbf{P}\beta_g)^T(u_g - \mathbf{P}\beta_g)$ with respect to β_g . That is, we find a vector of coefficients b_g which yields a vector $\mathbf{P}b_g$ that is close to u_g . The stacked regression rule classifies x_0 according to the rule

$$\eta^{\text{SR}}(x_0) = \arg \max_g P(x_0)b_g,$$

where x_0 is an unclassified observation and $P(x_0)$ is the row vector of class probability estimated obtained from the r classifiers. In practice, some care must be taken in finding b_g . For example, $\mathbf{P}^T\mathbf{P}$ is not full rank; we use the Moore-Penrose inverse of $\mathbf{P}^T\mathbf{P}$ in solving for b_g . Additionally, a backwards selection algorithm was used to eliminate those columns of \mathbf{P} that were not judged to be useful. LeBlanc and Tibshirani (1996) provide additional details.

4.2 Mojirsheibani's Method

Mojirsheibani (1999) proposed a substantially different approach to combining multiple classification rules. Given an unclassified observation x_0 , and classifiers η_1, \dots, η_r , Mojirsheibani's method identifies the subset $C_0 \subset X_n$ of observations that are classified identically to x_0 by all r classifiers, i.e.,

$$C_0 = \{x_i \in X_n \mid \eta_l(x_i) = \eta_l(x_0), l = 1, \dots, r\}.$$

Mojirsheibani's classifier assigns x_0 to the plurality group among C_0 , i.e.,

$$\eta^{\text{M}}(x_0) = \arg \max_g \#\{x_i \in C_0 \mid y_i = g\},$$

where $\#A$ is the cardinality of the set A .

4.3 The Product Rule

The plug-in Bayes rule of Section 3 can be viewed as a method of combining two classifiers; in that case, it combines the MID spatial classifier with another classifier such as k -NN. A generalization for combining r classifiers defines the *product rule* as

$$\eta^{\text{PROD}}(x_0) = \arg \max_g \frac{\prod_{v=1}^r P_g^v(x_0)}{\sum_{j=1}^c \prod_{v=1}^r P_j^v(x_0)},$$

where $P_j^l(x_0)$ is the estimated probability of membership in class j obtained from the l th of r classifiers (Steele 2000). Of course, one of these classifiers could be the MID spatial classifier.

4.4 Bootstrap Aggregation

Bootstrap aggregation, or bagging, is fundamentally different from the previously discussed methods. The aim of bootstrap aggregation is to reduce the variance of a classifier by averaging many bootstrap versions of the classifier (Hastie et al. 2001, Ch. 8). This is accomplished by drawing B bootstrap samples from X_n , using each sample to create a new classifier η^{*b} , $b = 1, \dots, B$, and aggregating the results. Aggregation may be accomplished by assigning x_0 to the plurality group among the predictions $\eta^{*1}(x_0), \dots, \eta^{*B}(x_0)$. Alternatively, if $P_g^{*b}(x_0)$ is the class g probability estimate obtained from the b th bootstrap classifier, then $P_g^{*b}(x_0)$ can be estimated by the average $P_g^{\text{BA}}(x_0) = B^{-1} \sum_{b=1}^B P_g^{*b}(x_0)$, and the bagging classifier can be defined according to $\eta^{\text{BA}}(x_0) = \arg \max_g P_g^{\text{BA}}(x_0)$.

The estimator $P_g^{\text{BA}}(x_0)$ is a Monte Carlo approximation to an exact bootstrap aggregation classifier $E_{\hat{F}} P_g(x_0)$, where $P_g(x_0)$ is the class g membership probability estimator, and expectation is with respect to the empirical distribution function \hat{F} placing probability mass n^{-1} at each $x_i \in X_n$. In almost all cases, $E_{\hat{F}} P_g(x_0)$ is intractable. However, if the k -NN classifier is used to compute $P_g(x_0)$, then $E_{\hat{F}} P_g(x_0) = E_{\hat{F}} P_g^{\text{kNN}}(x_0)$ can be computed analytically, and bootstrap sampling is not necessary. The analytical calculation begins with the expansion

$$E_{\hat{F}} P_g^{\text{kNN}}(x_0) = E_{\hat{F}} \frac{1}{k} \sum_{j=1}^k \Psi(y_{0,j} = g). \quad (6)$$

Calculation of the right-hand side of (6), requires $E_{\hat{F}} \Psi(y_{0,j} = g)$, $j = 1, \dots, n$. This term is not difficult because there are only n possible j th closest neighbors to t_0 . Specifically, let $t_{0,j}^*$ denote the j th closest covariate vector to t_0 among a bootstrap sample $T_n^* = \{t_1^*, \dots, t_n^*\}$ drawn randomly and with replacement from $T_n = \{t_1, \dots, t_n\}$. Let $P_{\hat{F}}(t_{0,j}^* = t_{0,i})$ denote the probability that $t_{i,0}$, the i th closest covariate vector to t_0 among T_n , is the j th closest among T_n^* . Then,

$$E_{\hat{F}} \Psi(y_{0,j} = g) = \sum_{i=1}^n P_{\hat{F}}(t_{0,j}^* = t_{0,i}) \Psi(y_{0,i} = g),$$

and

$$E_{\hat{F}} P_g^{kNN}(x_0) = \frac{1}{k} \sum_{j=1}^k \sum_{i=1}^n P_{\hat{F}}(t_{0,j}^* = t_{0,i}) \Psi(y_{0,i} = g).$$

To calculate $P_{\hat{F}}(t_{0,j}^* = t_{0,i})$, note that $t_{0,j}^* = t_{0,i}$ will occur if T_n^* contains at most $j-1$ copies of $t_{0,1}, \dots, t_{0,i-1}$ and at least j copies of $t_{0,1}, \dots, t_{0,i}$. If a ranges over the number of copies of $t_{0,1}, \dots, t_{0,i-1}$ and is constrained to $\{0, \dots, j-1\}$, and b ranges over the number of copies of $t_{0,i}$ while constrained to $\{j-a, \dots, n-a\}$, then there must be $n-a-b$ copies of $t_{0,i+1}, \dots, t_{0,n}$ to fill out T_n^* . Hence,

$$P_{\hat{F}}(t_{0,j}^* = t_{0,i}) = \frac{1}{n^n} \sum_{a=0}^{j-1} \sum_{b=j-a}^{n-a} \binom{n}{a} (i-1)^a \binom{n-a}{b} (n-i)^{n-a-b}.$$

The exact bootstrap aggregation k -NN classifier was previously developed by Steele and Patterson (2000) from a different standpoint: that of replacing the discrete outcomes $\Psi(y_{0,i} = g)$ in formula (1) with smoothed versions. The smoothed versions were the resampling expectations $E_{\hat{F}} \Psi(y_{0,j} = g)$. They present simulation examples showing the mean square error of the estimator $E_{\hat{F}} P_g^{kNN}(x_0)$ being smaller than $P_g^{kNN}(x_0)$, and error rates for the exact bootstrap aggregation k -NN classifier that are smaller than the ordinary k -NN classifier, and the distance-weighted k -NN classifiers of Dudani (1976), and Macleod et al. (1987).

4.5 Boosting

Unlike bagging, the objective of boosting is not produce a classifier with reduced variance compared to the base classifier, but to generate a more accurate classifier. As with bagging, B samples are drawn from X_n , though the probabilities of drawing each observation are modified so that hard-to-classify observations are more likely to be sampled. Classification is accomplished via a weighted voting procedure where the weight given to the b th classifier is determined by how well it classifies the hard-to-classify observations. There is a plethora of boosting methods; we used Ada-BoostM.1 or Discrete Ada-Boost (Freund and Schapire 1997; Friedman et al. 2000).

5. Data

Four Landsat TM scenes, Path 39, Row 27 (P39/R27), P39/R28, P40/R27, and P40/R28, were used to compare combination and ensemble classifiers. These scenes were mapped in collaboration with the USDA Forest Service, Northern Region, as part of a larger project to construct databases on land cover, forest canopy closure, and tree size class for nine Landsat TM scenes covering central Montana. The five National Forests with land management authority in this area were responsible for providing training samples. Land cover mapping and database construction were carried out by the Wildlife Spatial Analysis Laboratory, Montana Cooperative Wildlife Research Unit, located at the University of Montana. These Landsat scenes are located in a transition zone between the northern Great Plains and the Rocky Mountains. Consequently, each scene is spatially variable with respect to topography, climate and vegetation. From an areal extent, nearly all forest lands are located in the mountains, and these forests are exclusively conifer with the exception of small, sporadic, aspen forests. Approximately 50% of the combined

scene area is comprised of xeric grass and sagebrush cover types; another 28% is forest, 18% agriculture, and 1% urban.

Polygon map units were formed using a two-stage, digital classification process developed by Ma et al. (2001). In the first stage, land cover patterns were delineated by unsupervised classification using the ISODATA routine in the Erdas Imagine Software, followed by image segmentation. The latter step involved a rule and object-based process (Ford et al., 1997) which combined adjacent pixels of the same spectral class into contiguous areas greater than or equal to a user-designated minimum map unit size. These spatial units constitute the map polygons; the training samples are a subset of these polygons.

Accuracy was estimated using 10-fold cross-validation (Efron and Tibshirani 1993, Ch. 17). A k -fold cross-validation splits the data into k disjoint subsets of approximately equal size. The b th test subset, E_b , $b = 1, \dots, k$, is classified using training set $T_b = \{x_i \in X_n \mid x_i \notin E_b\}$, and the predictions are compared to the recorded memberships to obtain accuracy estimates.

6. Results

The EB 10-NN classifier tended to yield slightly greater accuracy estimates than the 10-NN classifier (Table 1).

Table 1. Number of classes (c), number of observations (n), and 10-fold cross-validation accuracy estimates (% correctly classified) for the four Landsat TM scene data sets.

Landsat TM scene	c	n	10-NN	EB 10-NN
P39/R27	17	2446	62.8	63.0
P39/R28	18	4242	58.6	59.0
P40/R27	19	2995	61.4	61.9
P40/R28	16	3013	66.5	66.4

The product rule, stacked regression and Mojirsheibani's method were used to combine the EB 10-NN classifier and tree classifiers (EB 10-NN+Tree), the EB 10-NN classifier and MID classifier, (EB 10-NN+MID), and tree and MID classifiers, (Tree+MID). As there were four data sets, the total number of 10-fold cross-validation estimates of accuracy was $3 \times 3 \times 4 = 36$. These are shown in Figure 1. The performance of the EB 10-NN+MID, Tree+MID combinations were similar: the largest estimates were obtained from the product rule, followed by stacked regression, and lastly, Mojirsheibani's method. For the EB 10-NN+Tree classifier, stacked regression yielded the largest accuracy estimates for all 4 data sets and neither the product rule nor Mojirsheibani's method produced consistently larger estimates than the other.

A combination classifier can be bagged or boosted because ensemble methods generate new versions of a classifier and classify by weighting the votes of each classifier. However, in this study it was only practical to bag and boost the product rule combination classifiers because the computational demands of stacked regression and Mojirsheibani's method were too great. Figure 2 shows that the estimated accuracy of the EB 10-NN classifier did not improve from bagging or boosting, as should be expected, because this classifier is itself the exact bagging version of the 10-NN classifier. The MID spatial classifier was not improved by either bagging or boosting. Bagging and boosting improved the tree classifier, and product rule combinations involving the tree classifier

(i.e., the Tree+MID and EB 10-NN+Tree classifiers). Generally, there is little evidence that boosting is more effective than bagging for these data sets.

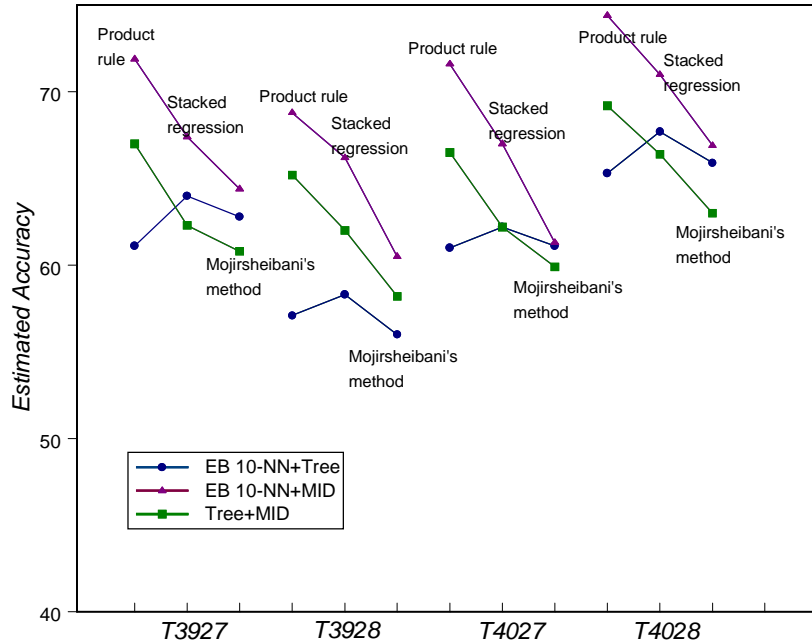


Figure 1. Tenfold cross-validation accuracy estimates (%) obtained from the combination methods. Results are organized by data set along the x-axis, and by combination method within data set. Within each data set, the product rule results are leftmost, the stacked regression results are in the center, and results from Mojirsheibani's method are rightmost.

7. Discussion

Our results differ from most comparative studies on bagging and boosting (e.g., Friedman et al. 2000, Opitz and Maclin 1999) in several respects. Our results do not conform to the conventional wisdom that Ada-Boosting of a tree classifier is a best method. For these data sets, the 10-NN+MID and EB 10-NN+MID classifiers were more accurate than the tree classifier, and neither bagging nor boosting of the tree classifier provides enough improvement to beat these spatial classifiers. We believe this is attributable to the value of spatial information. Though the MID classifier alone is not accurate, its information is quite different than that carried by conventional covariates (primarily, spectral reflectance variables). Surprisingly, the simple plug-in version of Bayes rule outperformed stacked regression and Mojirsheibani's method.

An advantage of tree classifiers is that very different covariates (e.g., continuous and categorical) may be used without problems of mismatched scales arising. In this case, though, the covariates are measured on very similar scales, and scale differences are probably not degrading the use of Euclidean distance as a measure of distance between observations (and groups). Because Euclidean distance is the basis of the 10-NN and EB

10-NN classifiers, we believe this to explain why they perform as well, or better than the tree classifier.

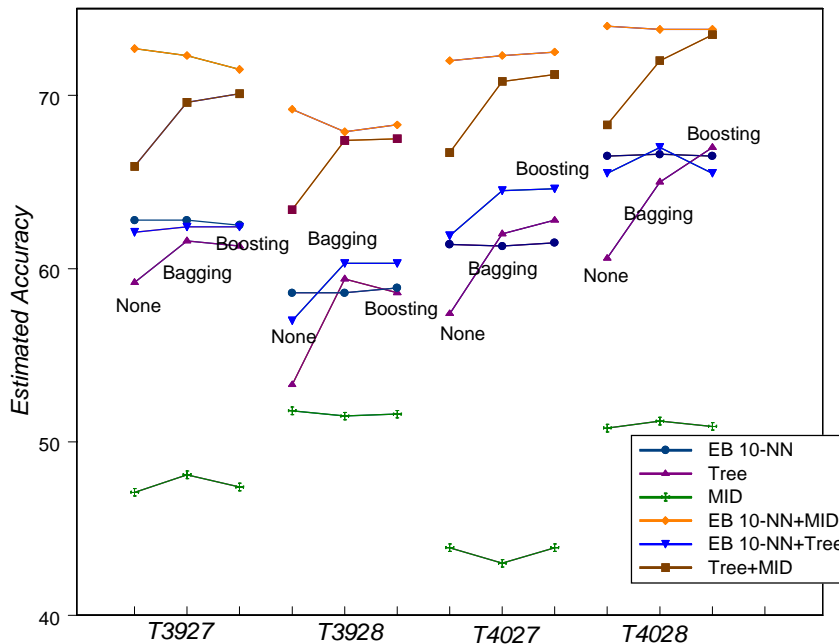


Figure 2. Accuracy estimates for the ensemble classifiers. Results are organized by data set along the x-axis, and by ensemble method within data set. Within each data set, the leftmost results are without bagging or boosting, the bagging results are in the center, and the rightmost results were obtained by boosting.

References

- Breiman, L. 1996. Stacked regressions. *Machine Learning*, **24**, 51-64.
- Breiman, L., Friedman, J., Olshen, R. and Stone, C. 1984. *Classification and Regression Trees*. Wadsworth.
- Carpenter, G.A., Gopal, S., Macomber, S., Martens, S., Woodcock, C.E., and Franklin, J. 1999. A neural network method for efficient vegetation mapping. *Remote Sensing of Environment*, **70**, 326-338.
- Dudani, S.A. 1976. The distance-weighted k -nearest-neighbor rules. *IEEE Transactions on Systems, Man, and Cybernetics*, **8**, 311-313.
- Efron, B. and Tibshirani, R.J. 1993. *An Introduction to the Bootstrap*. Chapman and Hall.
- Ford, R., Ma, Z., Barsness, S., and Redmond, R.L. 1997. Rule-based aggregation of classified imagery. *Proceedings of the 1997 ACSM/ASPRS Annual Convention*,

Technical Papers Volume 3, Remote Sensing & Photogrammetry. American Society for Photogrammetry and Remote Sensing, Bethesda, MD, USA, pp. 115-123.

Friedman, J., Hastie, T. and Tibshirani, R. 2000. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, **28**, 337-407.

Freund, Y. 1995. Boosting a weak learning algorithm by majority. *Information and Computation*, **121**, 256-285.

Freund, Y. and Schapire, R.E. 1996. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, Morgan Kaufman, San Francisco, pp. 148-156.

Freund, Y. and Schapire, R.E. 1997. A decision-theoretic generalization of online learning and application to boosting. *Journal of Computer and System Sciences*, **55**, 119-139.

Hastie, T., Tibshirani, R. and Friedman, J. 2001. *The Elements of Statistical Learning*. Springer.

Kartikeyan, B., Gopalakrishna, B., Kalaburme, M.H., and Majumdar, K.L. 1994. Contextual techniques for classification of high and low resolution remote sensing data. *International Journal of Remote Sensing*, **15**, 1037-1051.

LeBlanc, M. and Tibshirani, R. 1996. Combining estimates in regression and classification. *Journal of the American Statistical Association*, **91**, 1641-1650.

Lee, T.C.M. 2000. A minimum description length-based image segmentation procedure, and its comparison with a cross-validation-based segmentation procedure. *Journal of the American Statistical Association*, **95**, 259-270.

Ma, Z., Redmond, R.L., and Hart, M. 2001. Mapping vegetation across large geographic areas: integration of remote sensing and GIS to classify multisource data. *Photogrammetric Engineering and Remote Sensing*, **67**, 295-307.

Macleod, J.E.S., Luk, A. and Titterton, D.M. 1987. A re-examination of the distance-weighted k -nearest-neighbor classification rule. *IEEE Transactions on Systems, Man, and Cybernetics*, **17**, 689-696.

McLachlan, G.J. 1992. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley.

Mojirsheibani, M. 1999. Combining classifiers via discretization. *Journal of the American Statistical Association*, **94**, 600-609.

Optiz D. and Maclin, R. 1999. Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research*, **11**, 169-198.

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

Schapire, R.E. 1990. The strength of weak learnability. *Machine Learning*, **5**, 197-227.

Sharma, K.M.S. and Sarkar, A. 1998. A modified contextual classification technique for remote sensing data. *Photogrammetric Engineering and Remote Sensing*, **64**, 273-280.

Steele, B.M. 2000. Combining multiple classifiers: An application using spatial and remotely sensed information for land cover type mapping. *Remote Sensing of Environment*, **74**, 545-556.

Steele, B. M. and Patterson, D.A. 2000. Ideal bootstrap estimation of expected prediction error for k -nearest neighbor classifiers: Applications for classification and error assessment. *Statistics and Computing*, **10**, 349-355.

Steele, B.M., and Redmond, R.L. 2001. A method of exploiting spatial information for improving classification rules: Application to the construction of polygon-based land cover maps. *International Journal of Remote Sensing*, **22**, 3143-3166.

Stuckens, J., Coppin, P.R., and Bauer, M.E. 2000. Integrating contextual information with per-pixel classification for improved land cover classification. *Remote Sensing of Environment*, **71**, 282-296.

Van Deusen, P.C. 1995. Modified highest confidence first classification. *Photogrammetric Engineering and Remote Sensing*, **61**, 419-425.

Watson, D.F. and Philip, G.M. 1985. A refinement of inverse distance weighted interpolation. *Geo-Processing*, **2**, 315-327.

Wolpert, D. 1992. Stacked generalization. *Neural Networks*, **5**, 241-259.