Supervised Image Classification by Contextual AdaBoost Based on Posteriors in Neighborhoods

Ryuei Nishii, Member, IEEE, and Shinto Eguchi

Abstract—AdaBoost, one of machine learning techniques, is employed for supervised classification of land-cover categories of geostatistical data. We introduce contextual classifiers based on neighboring pixels. First, posterior probabilities are calculated at all pixels. Then, averages of the posteriors are obtained in various neighborhoods, and they are used as contextual classification functions. Weights for the classification functions can be determined by minimizing the empirical risk with multiclass. Finally, a linear combination of classification functions is obtained. The proposed method is applied to artificial multispectral images and a benchmark data set. It shows an excellent performance similar to the MRF-based classifier with much less CPU time.

Index Terms—Bayes rule, machine learning, MRF, posterior probability, segmentation.

I. INTRODUCTION

IMAGE classification is an important issue of the remote sensing community as well as other communities. In the literature, statistical approach is much discussed [1–4], and [5] is a review paper on statistical pattern recognition. Fusion techniques and machine learning techniques are also discussed, see e.g. [6–8].

AdaBoost proposed by [9] is one of machine learning techniques and has been progressed for pattern recognition rapidly and widely in a paradigm of supervised learning. It combines weak classifiers into a strong one based on weighted voting system, and shows high performance in various fields, see [10] and [11]. Here, the term “a weak classifier” will be used in the context of learning. It means that the classifier is better than random guess a little.

In this paper, we introduce AdaBoost for contextual image classification for geostatistical data. First, we prepare a posterior probability distribution given feature vector at each pixel. The posteriors can be introduced by machine learning algorithms as well as statistical methods. Then, averages of the posteriors are calculated in different radii of neighborhoods, and we regard them as classification functions. The weights for the functions are obtained by minimizing an empirical risk defined for a problem of multiclass classification. Finally, a linearly combined function gives a contextual voting machine.

The proposed method is shown to be a very fast algorithm. In our empirical studies, the method needs only about 1/1000 of CPU time required by MRF-based classifiers implemented by [7]. The performance of our method keeps equivalent to that of MRF-based classifiers. Also, the proposed method is very flexible about incorporating contextual information into since the posteriors can be derived from various techniques: support vector machines (SVM), AdaBoost, and artificial neural networks, see e.g. [12–14].

In Section II, AdaBoost is briefly reviewed. A simple example illustrates AdaBoost with binary class. Then, real AdaBoost is explained in multiclass case. Section III defines contextual classifiers. Neighborhoods of a pixel and the averages of the posteriors there are defined, which will be made use of to build classifiers. The proposed method is applied to two data sets in Section IV. It is shown that the method needs less computation time and gives an excellent performance similar to the MRF-based classifier. Section V concludes the paper.

II. A SHORT REVIEW OF ADA BOOST WITH MULTICLASS

We will start this section by a simple example for illustration of AdaBoost.

A. AdaBoost with binary class

Suppose that a feature vector \( x \in \mathbb{R}^m \) is observed from one of two categories \( C_1 \) or \( C_{-1} \). Let \( g_k(x) \) be a classifier taking a value in a label set \( \{+1, -1\} \) for \( k = 1, 2, 3 \). If three classifiers are equally trustworthy, a new function \( \text{sign}(g_1(x) + g_2(x) + g_3(x)) \) is a boosted classifier based on majority vote, where \( \text{sign}(z) \) is the sign of the argument \( z \). If \( g_1 \) is most reliable, \( g_2 \) comes next and \( g_3 \) comes last, a classifier \( \text{sign}(c_1g_1(x) + c_2g_2(x) + c_3g_3(x)) \) would be a locally-better boosted classifier based on weighted vote, where \( c_1 > c_2 > c_3 \) are positive constants. They are tuned by minimizing the empirical risk which will be defined shortly.

In general, let \( F(x) \) be a classifier of a feature vector \( x \) with the true label \( y \in \{1, -1\} \). The label \( y \) is estimated by \( \text{sign}(F(x)) \). Then, AdaBoost takes the loss function \( \exp(-yF(x)) \). Let \( \{(x_i, y_i) | i = 1, 2, \ldots, n\} \) be a set of training data. A classification function \( F \) is selected by minimizing the empirical risk \( \sum_{i=1}^{n} \exp(-y_iF(x_i))/n \). In the example above, \( F(x) \) is set to \( c_1g_1(x) + c_2g_2(x) + c_3g_3(x) \) and the coefficients are tuned by minimizing the empirical risk. A fast sequential procedure for minimizing the risk is well-known, see e.g., [10].

AdaBoost aims to combine weak classifiers. A typical weak classifier is a decision stump defined by a function...
\[ \delta \text{ sign}(x^j - t), \text{ where } \delta = \pm 1, \ t \in \mathbb{R}, \text{ and } x^j \text{ denotes a } j\text{-th variate of the feature vector } x. \text{ Then, the poorly performed classification functions are going to learn training data, and this process finally builds a strong one.} \]

**B. Real AdaBoost with multiclass**

Next, consider AdaBoost in multiclass case. Let \( \mathcal{D} = \{1, \ldots, n\} \) be a training area consisting of \( n \) pixels, and each pixel is belonging to one of \( g \) possible categories \( C_1, \ldots, C_g \). Suppose that an \( m \)-dimensional feature vector \( x_i \in \mathbb{R}^m \) is observed at each pixel \( i \). A label of the category covering the pixel \( i \) is denoted by \( y_i \in \{1, \ldots, g\} \). Now, let a function \( f(x, k) \) be a (weak) classification function. We allocate a feature vector \( x \) into a category label by the following maximizer:

\[
\hat{y} = \arg \max_{k \in \{1, \ldots, g\}} f(x, k).
\] (1)

In the ordinary setting of AdaBoost, the weak classification function \( f \) is supposed to take only two values as the decision stump mentioned earlier. We also consider the classification function taking real values. Let \( p(x | k) \) be a class-conditional probability density function of the \( k\)-th category for \( k \in \{1, \ldots, g\} \). Then, a posterior probability:

\[
p(k | x) = \pi_k p(x | k) / \sum_{\ell=1}^g \pi_\ell p(x | \ell)
\] (2)
gives a strong classification function taking positive values, where \( \pi_\ell \) is a prior probability of the category \( C_\ell \). The posterior (2) is closely related to logistic-type discriminant functions, see [15].

AdaBoost aims to combine weak classifiers into a strong classifier by a weighted majority vote. Let \( \mathcal{F} = \{ f(x, k) \} \) be a set of weak classification functions. The loss of misclassification due to the classification function \( f \) is assessed by the following exponential loss function:

\[
L(x, k | y) = \exp[f(x, k) - f(x, y)]
\] (3)
for \( k \in \{1, \ldots, g\} \), where \( y \) is the true label of the feature vector \( x \). The loss function (3) is an extension of the exponential loss with binary class. The empirical risk is defined by

\[
R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^g \exp[f(x_i, k) - f(x_i, y_i)]
\] (4)

where \( \{(x_i, y_i)\} \) is a set of training data on the area \( \mathcal{D} \).

Let \( f \) and \( F \) be classification functions (fixed). Then, the optimal coefficient \( c \) which gives the minimum value of the empirical risk \( R_{\text{emp}}(F + cf) \) is denoted by \( c^* \):

\[
c^* = \arg \min_{c \in \mathbb{R}} \{R_{\text{emp}}(F + cf)\}.
\]

If the function \( f(\cdot, k) \) takes only two values, the optimal coefficient \( c^* \) can be expressed in the closed form. If \( f(\cdot, k) \) takes real values, there is no closed form to the coefficient \( c^* \). Then, we will take an iterative procedure for the estimation.

**III. NEIGHBORHOODS AND CONTEXTUAL CLASSIFIERS**

As we have seen in Section II, AdaBoost requires a set of weak classification functions defined in the feature space. In the sequel we will consider a set of classification functions for the contextual classification, and the functions are not confined to be weak.
Let \( d(i, j) \) be a distance between centers of two pixels \( i \) and \( j \) in the area \( \mathcal{D} \). Then, define a subset \( U_r(i) \) of \( \mathcal{D} \) with center \( i \) and radius \( r \) as
\[
U_r(i) = \{ j \in \mathcal{D} \mid d(i, j) = r \} \quad \text{for} \quad r = 0, 1, \sqrt{2}, 2, \ldots
\]
Note that the subset \( U_r(i) \) constitutes an isotropic ring region and is different to an ordinary neighborhood. See Fig. 1 for the subsets \( U_r(i) \) for \( r = 0, 1, \sqrt{2}, 2 \). It is seen that \( U_0(i) = \{ i \} \), \( U_1(i) \) is the first-order neighborhood of the pixel \( i \), and \( U_1(i) \cup U_2(i) \) forms the second-order neighborhood. Then, the posterior probability \( p(k \mid x) \) defined by (2) is a measure of confidence of the current classification. Such a measure is defined by machine learning techniques as well as statistical techniques, see [12-14].

Define the average of posterior probabilities in the subset \( U_r(i) \) by
\[
\bar{\rho}_r(k \mid i) = \frac{\sum_{j \in U_r(i)} p(k \mid x_j)}{|U_r(i)|} \quad \text{if} \quad |U_r(i)| > 0
\]
\[
\bar{\rho}_r(k \mid i) = 1 \quad \text{otherwise}
\]
for \( r = 0, 1, \sqrt{2}, \ldots \), where \(|S|\) denotes the cardinality of a set \( S \). Immediately, it holds that the averaged posterior \( \bar{\rho}_0(k \mid i) \) with radius \( r = 0 \) is equal to the posterior \( p(k \mid x) \). Hence, noncontextual classification is done by a strong classification function \( \bar{\rho}_0(k \mid i) \). If the spatial dependency between the categories exists, the averaged posteriors \( \bar{\rho}_1(k \mid i) \) in the adjacent pixels to the pixel \( i \) also may have information for classification. If the spatial dependency is stronger, \( \bar{\rho}_r(y \mid i) \) with larger \( r \) is also useful. Thus, we adopt the average of the posteriors \( \bar{\rho}_r(k \mid i) \) as a classification function of the center pixel \( i \).

Obviously, efficiency of the posteriors as classification functions would be intuitively in the following order:
\[
\bar{\rho}_0(k \mid i), \quad \bar{\rho}_1(k \mid i), \quad \bar{\rho}_{\sqrt{2}}(k \mid i), \quad \bar{\rho}_2(k \mid i), \quad \cdots
\]
The coefficients to the classifiers \( \bar{\rho}_r(\cdot \mid \cdot) \) can be tuned by minimizing the empirical risk (4) in the supervised image classification.

Possible candidates of contextual classification functions are as follows.

1. \( \bar{\rho}_r(k \mid i) \) in the subset \( U_r(i) \) given at (5), or log \( \bar{\rho}_r(k \mid i) \).
2. Thresholding of the function \( \bar{\rho}_r(k \mid i) \) by
\[
\bar{\rho}_r^+(k \mid i) = \begin{cases} 
1 & \text{if} \quad k = \text{argmax}_k \bar{\rho}_r(\ell \mid i), \\
0 & \text{otherwise}.
\end{cases}
\]
3. Proportion of the labeled pixels in \( U_r(i) \):
\[
\bar{\varphi}_r(k \mid i) = \begin{cases} 
\frac{|\{ j \in U_r(i) \mid y_j = k \}|}{|U_r(i)|} & \text{if} \quad |U_r(i)| > 0, \\
0 & \text{otherwise}.
\end{cases}
\]
4. Majority vote in \( U_r(i) \):
\[
\bar{q}_r^+(k \mid i) = \begin{cases} 
1 & \text{if} \quad k = \text{argmax}_k \bar{q}_r(\ell \mid i), \\
0 & \text{otherwise}.
\end{cases}
\]
We here note that all the classification functions in the above are well-defined for the training data. However, the functions given in (2a) and (2b) cannot be defined over the test data because test labels in the neighborhood are unknown. If tentative estimates due to some strong classifier are available, the functions (2a) and (2b) are utilized. But, the classifiers would supply only supplementary information.

We propose a following contextual classification procedure based on AdaBoost (Spatial AdaBoost), which is parallel to that established in the subsection B of Section II.

1) Let \( \mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_r \) be sets of (strong) classification functions defined by (1a) and (1b) based on various posteriors under consideration for a radius 0, 1, \sqrt{2}, \ldots, \text{r}
2) Find a classification function \( f \in \mathcal{F}_0 \) and a coefficient \( c \) which minimize the empirical risk \( R_{emp}(c, f) \) defined by the formula (4), say \( f_0 \) and \( c_0 \).
3) If the coefficient \( c_0 \) is negative, quit the procedure. Otherwise, consider the empirical risk \( R_{emp}(c_0, f_0 + c) \) with given \( c_0, f_0 \) in the previous step. Then, find the optimal classification function \( f \in \mathcal{F}_1 \) and coefficient \( c \) which minimize the empirical risk, say \( f_1 \) and \( c_1 \).
4) If \( c_1 \) is negative, quit the procedure. Otherwise, consider the empirical risk \( R_{emp}(c_0, f_0 + c_1, f_1 + c) \) for \( f \in \mathcal{F}_{\sqrt{2}} \). This is repeated and we obtain the final classification function \( F = c_0, f_0 + c_1, f_1 + \cdots + c_r f_r \).

Then, a test vector \( x_i \) is classified into a category with the label given by \( \text{argmax}_k F(x_i, k) \). Note that the classification function \( F(x_{i_r}) \) depends on the test data \( x_j \) satisfying the inequality \( d(i, j) \leq r \).

IV. NUMERICAL EXPERIMENTS

Our method is examined through two data sets; one is artificially generated and the other is a real data set. We put the set of classification functions by \( \mathcal{F}_r = \{ \log \bar{\rho}_r \} \) consisting of a single strong classification function based on Gaussian densities for \( r = 0, 1, \sqrt{2}, \ldots \) in both the cases.

A. Application to an artificial data set

The proposed method is applied to multispectral images generated over the image (a) of Fig. 2 with three categories \( q = 3 \). The labels 1, 2 and 3 correspond to the colors black, white and grey. The numbers of pixels from the categories are respectively given by 3330, 1371 and 3580. We simulate four-dimensional spectral images \((m = 4)\) at each pixel of the true image (a) following multivariate Gaussian distributions independently with mean vectors \( \mu(1) = (0 0 0 0)_T, \mu(2) = (1 1 0 0)^T/\sqrt{2}, \mu(3) = (1.0498 -0.6379 0 0)^T \) and common covariance matrix \( \sigma^2 I \), where \( I \) stands for the identity matrix. Test data are similarly generated over the same image (a),
and the Gaussian density functions are used for deriving the posteriors.

Considering the intuitive efficiency order (6) of the averages, we apply the following classification functions:

\[ c_0 \log \hat{p}_0, c_0 \log \hat{p}_0 + c_1 \log \hat{p}_1, \ldots, c_0 \log \hat{p}_0 + c_1 \log \hat{p}_1 + \cdots + c_{\sqrt{N}} \log \hat{p}_{\sqrt{N}} \]

to test data, where the coefficient \( c_0, c_1, \ldots, c_r \) are sequentially tuned by minimizing the empirical risk (4).

Fig. 3 plots the coefficients \( c_0, c_1, \ldots, c_{\sqrt{N}} \) against the squared radius \( r^2 \) in the case \( \sigma^2 = 1 \). Magnitude of the coefficients shows a reliability of the corresponding classifiers. The coefficient \( c_0 \) is not large, and this is ascertained by the fact that the high error rate due to \( \hat{p}_0 \) (the noncontextual classification function) is 41.75%.

Fig. 4 shows training and test error rates due to the functions \( c_0 \log \hat{p}_0 + c_1 \log \hat{p}_1 + \cdots + c_r \log \hat{p}_r \) for \( r = 0, 1, \ldots, \sqrt{50} \) in the case \( \sigma^2 = 1 \). Fig. 4 (a) shows that the neighborhood information reduces the test error rate monotonously, and the error rate is stable even if the radius \( r \) is too large, whereas (b) implies that the test error takes the minimum value at the radius \( r = \sqrt{13} \).

Fig. 2 (b) is the classified image obtained by the posterior \( \log \hat{p}_0 \), which is equivalent to the linear discriminant function. The classified images are getting closer to the true label (a) as the radius \( r \) is getting larger. Fig. 2 (f) shows the best result, see Table I.

Table I compares error rates due to Gaussian MRF-based (GMRF) and the proposed classifiers in two cases: \( \sigma^2 = 1, 4 \). Each row is corresponding to GMRF with the neighborhood system \( U_i(\phi) \cup U_i(\sqrt{2}\phi) \cup \cdots \cup U_i(\phi) \) and the classification function \( c_0 \log \hat{p}_0 + c_1 \log \hat{p}_1 + \cdots + c_r \log \hat{p}_r \). The third and fifth columns show CPU times (s) required by GMRF and the proposed method respectively. GMRF needs much computation time for tuning the dependency parameter [7], whereas the proposed method determines the coefficients sequentially with several iterations. It is shown that the proposed method is very fast procedure compared with the ordinary MRF-based classifier, and that both the classifiers show similar performance.
B. Application to a benchmark data set

Next, the proposed method is applied to a benchmark data set grss_df0006 for supervised image classification. The set, accessible by IEEE GRS-S Data Fusion reference database [16], consists of samples acquired by ATM and SAR (6 and 9 bands, respectively) with five agricultural categories in the Feltwell (UK) region. The numbers of the training and the test data are respectively 5072 and 5760 in a rectangular region of size $350 \times 250$.

In this case, the labels of the test and training data are observed sparsely. Note that the classification based on the radius $0$ means the noncontextual classification by the linear discriminant function. Table II tabulates error rates and CPU time due to GMRF obtained by [7] and the proposed method. The table draws a similar conclusion obtained by Table I.

Fig. 5 plots estimated coefficients for $\log \bar{p}_r$ against $r^2$. The estimated coefficient with $r^2 = 72$ is $-0.012975$, a negative value. This implies that the averaged posterior probability $\bar{p}_{\sqrt{72}}$ is not appropriate for the classification. Hence we must stop to combine the function with radius $\sqrt{72}$ or higher.

The Gaussian distribution with common covariance matrix is employed for deriving posterior probabilities, and the proposed method shows an excellent performance. If we fit Gaussian distributions with class-specific covariance matrices to the same data, the error rate is $14.7\%$. The coefficient to $\log \bar{p}_0$, however, turns to be a negative value. This implies that the exponential loss defined by (3) is too sensitive to outlying data. See [17] for a robust classification.

V. CONCLUSION

Real AdaBoost is introduced to provide contextual image classification. We define the classifiers based on posteriors on subsets of neighbors, and combine them by the AdaBoost-based method. Features of the proposed classification method are as follows.

1) Various forms of posteriors can be implemented into the method.
2) Various forms of neighborhoods can be also implemented.
3) The method requires much less computation time and shows similar performance to MRF-based classifiers.

1): We adopted the strong classification function by the posterior probability $p(k\mid x)$ defined by the formula (2). Hence, our method can be illustrated as “AdaBoost + Statistics”. The posterior, however, can be defined by SVM or AdaBoost. Hence, classification methods “AdaBoost + SVM” and “AdaBoost + AdaBoost” are also proposed. See [18] for the derivation of the posteriors in multiclass are calculated by the pairwise coupling of binary classifications.

2): We adopted isotropic neighborhoods of a center pixel (recall Fig. 1). Similarly, we can also employ directed neighborhoods, see Fig. 6. This means that the proposed method would be also applicable for classification of textured images.
Selection of forms of posteriors and neighborhoods is a new problem. Selection of the number of classifiers, or equivalently, determination of the range of the radius is also an old and new problem. Further, a robust loss function in multiclass must be studied.

ACKNOWLEDGMENT

The data set grss_dfc_0006 is provided by IEEE GRS-S Data Fusion Committee.

REFERENCES


