An Efficient Recursive Partitioning Algorithm for Classification, Using Wavelets

Vittorio Castelli
System Theory and Analysis dept.
IBM T.J. Watson Res. Ctr.
P.O. Box 218
Yorktown Heights, NY 10598
vittorio@us.ibm.com

Ioannis Kontoyiannis
Div of Applied Math & CS Dept
Brown University
182 George Street
Providence, RI 02906
yiannis@dam.brown.edu

August 1999

Abstract

We describe and analyze a new dyadic recursive partitioning algorithm for efficient classification of large two-dimensional data sets, called progressive classification. It uses generic (parametric or nonparametric) classifiers on a low-resolution representation of the data obtained using the discrete wavelet transform. In this representation, each point corresponds to a block of samples from the original data. At each step of the classification process, the algorithm either decides to classify the whole block as belonging to a certain class, or to re-examine the data at a higher-resolution level. We present simple theoretical results showing that, compared to sample-by-sample algorithms, progressive classification is computationally more efficient and also (under certain conditions) more accurate.

We outline how progressive classification deals with data in one dimension and in dimensions higher than three, and we briefly discuss the complexity/accuracy tradeoff.

1 Introduction

We describe and analyze a new method for performing efficient classification on large two-dimensional data sets ("images"). We apply a dyadic, recursive partitioning algorithm which generates a tree-structured subdivision of the data, using an adaptive rule based on the wavelet transform. For our purposes, an image is a real- or vector-valued configuration on a rectangular subset of the integer lattice $\mathbf{Z}^2$. With each point on the lattice ("pixel") we associate a real number (or a vector in $\mathbf{R}^d$) denoting its *pixel-value*, and a *label* denoting its *class*. The problem of classification consists of observing an image with known pixel-values but unknown labels and assigning a label to each pixel.

The application illustrated in the following example was the major motivation for our study; see [4, 22] for more details. Consider a database containing a large number of satellite images of the earth's surface, and suppose this database is constantly growing as newly acquired data is being added. The practical value and usability of the database depends very strongly on whether or not the images are classified during the pre-processing stage. Among other advantages, this allows users to perform content-based searches in the database [23, 15], and makes it possible to conduct global change studies on the classified data rather than on the original images.

The standard approach up to now (see Chettri *et al.* [6], Richards [17], Chettri and Cromp [5], Salu and Tilton [20], Paola and Schowengerdt [16] and the references therein) has been to examine the pixels one-by-one and classify them independently of the neighboring pixels. In practice, this approach suffers from two obvious defects. It is computationally very expensive, and (other than through smoothing) it does not take into account the strong correlations between the labels of neighboring pixels that occur in natural images. More complex methods that exploit correlations between labels and between pixel values (for instance, by using random fields to characterize both the label field and the reflectance process and by putting a Bayesian prior on the parameter space on the model), are generally computationally infeasible, due to the size and number of images in a typical satellite image database. No relief seems to be forthcoming in the near future, as the amount of satellite data acquired is increasing at a pace much faster than the corresponding increases in computing speed. See [22] for an extensive discussion.

Data on the one-dimensional and three-dimensional lattices are somewhat less common. An example of one-dimensional lattice data is the set of oil well measurements collected by lowering a package of instruments to the bottom of the well, and slowly pulling them back to the surface. The instruments measure several properties of the rocks surrounding the borehole, such as electrical resistivity, sonic velocity, gamma rays, induced radioactivity etc. Typically, as the instrument package is retrieved, it produces a set of measurements every 6 inches, which can be represented as a vector-valued random variable on the one-dimensional lattice. The measurements are then used to identify the bulk lithologies (rock types) encountered in the borehole via a classification process (for more details, see, for example, [14]). Three-dimensional data include atmospheric measurements, seismic data (used in the petroleum industry) and some types of medical images.

In this paper, we present a new statistical method for classification called *progressive classification*, and we argue that it can be used to classify transformed data quickly and accurately. The main gist of our approach is to look at a lower-resolution representation of the data, where each pixel corresponds to a k^d pixel-block from the original $d-$dimensional data set. At each step of the classification process the algorithm either assigns the same class label to the whole block, or it takes a "progressive step": it looks at a higher-resolution version of the data, where the original k^d pixel-block is now represented by 2^d pixels instead of just one, and examines each one of those 2^d pixels independently. The same process is repeated iteratively.

The low-resolution representation of the data is obtained by taking its Discrete Wavelet Transform (DWT) up L levels, and examining the lowest frequency subband.[1] Starting with an $M \times N$ pixel image, going up L levels reduces the size of the image by a factor of 4^L so that, starting classification at the lowest frequency subband, only $(MN)/4^L$ pixels have to be examined. Classifying a pixel at level ℓ (for some $\ell \leq L$) means that the same label is assigned to all the corresponding $2^\ell \times 2^\ell$ pixels in the original image. Taking the progressive step, on the other hand, means that the four pixels at level $(\ell - 1)$, corresponding to the one pixel we started with, are now examined independently, by looking at the lowest frequency subband of the DWT of the image at level $(\ell - 1)$. It is important to note that the choice of the classifier used within each level to decide on the classification of the whole block or the progressive step is completely arbitrary (we have implemented progressive versions of CART [3], nearest neighbor [7], learning vector quantization, maximum likelihood and neural network classifiers).

In practice, progressive classification was successfully implemented by Castelli *et al.* [4], where practical issues arising in the implementation were discussed, and extensive numerical results were presented to demonstrate its performance on large satellite images of the earth's surface. The purpose of this paper is to present a theoretical analysis that supports and helps interpret those results. From this analysis (presented in Sections 2 and 3), it is seen that, first, progressive classification achieves a significant speedup over pixel-by-pixel classification methods (which is not surprising), but also, for images with label values that are highly correlated, the progressive classifier will give more accurate

[1] The DWT is at the base of numerous algorithms in statistical signal processing, data compression, and image analysis; see the recent collection Antoniadis and Oppenheim [1], and the special issue Daubechies *et al.* [9]. The use of the DWT is partly motivated by the fact that, in a large number of applications, images are stored in a compressed format and the compression is based on the DWT.

results than the corresponding non-progressive classifier. The reason for the speed-up is clear; looking at a lower-resolution representation of the data and classifying (whenever possible) whole blocks, means that the total number of pixels examined will be often much smaller than MN. The reason for the improvement in classification accuracy is that the DWT produces a weighted average of the values from each $2^\ell \times 2^\ell$ block, so that the classification algorithm often tends to assume more uniformity in the data than may appear when looking at individual pixels.

The use of wavelets in classification has been proposed by several authors in the literature, where, unlike in the present approach, the wavelet transform is used to extract relevant features that can be used as predictors for the data (see, for example, Tate, Watson and Eglen [21]). A work closer in spirit to the present paper is Donoho [11], where anisotropic Haar bases are used to partition the observation space into classification regions in a way that is near-optimal with respect to minimax risk. In contrast, we are interested in both the error rate of the classifier and the classification complexity. In our case, the partition regions produced by the progressive classifiers are generated by a spatially adaptive blocking technique, and they depend on the choice of the wavelet basis as well as on the underlying classifier.

In Section 2, after some preliminaries on classification and wavelets we define the progressive classifier and present our analysis of progressive classifiers applied to one-dimensional data. These simple results provide some insight on the the more complex analysis for two-dimensional data developed in Section 3. Section 4.2 briefly shows how progressive classification can be applied to general lattice data in dimensions higher than two. In Section 4.1 we discuss the tradeoff between classification complexity and accuracy, and in Section 4.3 we discuss the choice of the wavelet basis. Section 5 shows the results of a small experiment comparing the performance of the progressive classifier, the corresponding baseline classifier and a Bayesian method known as Iterated Conditional Modes. Appendix A contains the proofs of the results in Section 2.

Throughout the paper, random variables are denoted by upper-case letters and their realizations by the corresponding lower-case letters. We write $\phi(\cdot)$ for the standard Normal density. Vectors are in lower-case boldface type, and vector-valued random variables are in upper-case boldface type. For $i \in \{0, 1\}$, we write $\bar{i} = 1 - i$, and $\delta_{i,j} = 1$ if $i = j$, $\delta_{i,j} = 0$ otherwise.

2 Classification in One Dimension

2.1 Preliminaries

First we give a brief description of the one dimensional DWT; Daubechies [8] has a complete account. Let $\mathbf{x} = \{x_i \ ; \ i = 0, 1, \ldots, 2^L - 1\}$ be the data sequence to be transformed, and fix two wavelet (finite impulse response) filters $\mathbf{F}$ and $\mathbf{G}$, where by a "filter" we mean a (typically short) sequence of real numbers, and "wavelets" correspond to a class of regular filters that are related to bases in $L^2(\mathbf{R})$; see Daubechies [8], Rioul and Vetterli [19], and Rioul [18]. The DWT consists of the following recursive operation. The sequence $\mathbf{x}$ is first filtered through $\mathbf{F}$ and $\mathbf{G}$ separately, and the results are downsampled to obtain two new sequences $\mathbf{v}^1 = \{v_i^1\}$ and $\mathbf{w}^1 = \{w_i^1\}$ (where "filtering" means "convolution" and "downsampling" means discarding every other sample). We call $\mathbf{v}^1$ and $\mathbf{w}^1$ the *subbands of the DWT of* $\mathbf{x}$ *at level 1*, and their elements are the *level-1 wavelet coefficients of* $\mathbf{x}$. To take the DWT up L levels we repeat the described process recursively on the sequence $\mathbf{w}^1$ another $(L-1)$ times, to obtain sequences $\mathbf{v}^2, \mathbf{v}^3, \ldots, \mathbf{v}^L, \mathbf{w}^L$, called the *subbands of* $\mathbf{x}$ *at levels* $2, 3, \ldots, L$; the sequence $\{w_i^L \ v_i^L \ \ldots v_i^1\}$ is called *the L-level DWT of* $\mathbf{x}$, and its elements are the *wavelet coefficients.* The sequences $\{w_i^1\}$, $\{w_i^2\}, \ldots, \{w_i^L\}$ are the *multiresolution approximations of* $\mathbf{x}$ *at levels* $1, 2, \ldots, L$, respectively, and together with $\mathbf{x} = \{w_i^0\}$ they form a *multiresolution pyramid.* Recall that the DWT is reversible in that the data can be progressively reconstructed to any required level of resolution from their wavelet coefficients.

For simplicity, most of our analysis will be based on using Haar wavelets, that is, the DWT will be specified by the filters $\mathbf{G} = (1/\sqrt{2}, 1/\sqrt{2})$ and $\mathbf{F} = (1/\sqrt{2}, -1/\sqrt{2})$. The main results of this paper, however, hold for general wavelet filters. In the case of Haar wavelets, the level-ℓ wavelet coefficients $\{w_i^\ell\}$ will be scaled averages of distinct subsequences from $\mathbf{x}$:

$$w_i^\ell = \frac{w_{2i}^{\ell-1}}{\sqrt{2}} + \frac{w_{2i+1}^{\ell-1}}{\sqrt{2}} = 2^{-\ell/2} \sum_{j=0}^{2^\ell - 1} x_{2^\ell i + j} = 2^{\ell/2} \left(\frac{1}{2^\ell} \sum_{j=0}^{2^\ell - 1} x_{2^\ell i + j} \right). \tag{1}$$

We refer to the coefficients $\{w_i^\ell\}$ as the *wavelet coefficients from the lowest frequency subband of the ℓ-level wavelet transform of* $\mathbf{x}$, or, for short, the *level-ℓ wavelet coefficients of* $\mathbf{x}$. (In the wavelet literature, the coefficients $\{w_i^\ell\}$ are often referred to as the "scaling function coefficients" or the "approximation coefficients.")

Next a few words about classification. Let x be an observation (for instance, the vector of reflectance values at a particular location of a remotely-sensed multispectral image), and let θ describe an associated state of nature (for instance, the land-cover class of the region corresponding to the pixel), usually referred to as the "label" of the observation. The pair (x, θ) is called a labeled sample.

For the sake of simplicity, in the rest of the paper θ will take one of two values, 0 and 1, but all the results easily extend to any finite number of different classes. The problem of classification consists of guessing the value of the label θ given x. The simplest statistical setting for the classification problem treats the pair (x, θ) as a realization of a pair of dependent random variables (X, Θ), where $\Pr(\Theta = 0) = \pi$, $\Pr(\Theta = 1) = 1 - \pi$, and X is conditionally distributed according to $F_\Theta(\cdot)$ given Θ. In general, we assume that $F_\Theta(\cdot)$ has density $f_\Theta(\cdot)$ with respect to an appropriate measure (we do not specify the reference measure in order to give a unified treatment of continuous and discrete observations, having densities with respect to Lebesgue measure and counting measure, respectively). A classifier is simply a mapping $\hat{\theta}$ from the observation space to the set $\{0, 1\}$, assigning a label $\hat{\theta}(x)$ to each observation x. The pre-images of 0 and 1 are here denoted by Π_0 and Π_1, so that $\hat{\theta}(x) = 0$ for each $x \in \Pi_0$, and $\hat{\theta}(x) = 1$ for each $x \in \Pi_0$.

Classification rules differ in the way they partition the observation space (see, for instance, Devroye et al. [10].) If the densities and the prior probabilities are known, then there is an optimal solution to the classification problem called the Bayes' decision rule, which labels the samples using the likelihood ratio $[\pi\ f_0(X)] \,/\, [(1 - \pi)\ f_1(X)]$. If the likelihood ratio is greater than one, the Bayes decision rule decides $\hat{\theta} = 0$, whereas it decides $\hat{\theta} = 1$ otherwise. The Bayes decision rule is the optimum classifier for independent and identically distributed (i.i.d.) data, in the sense that it minimizes the probability of classification error (see, e.g., Duda & Hart [12]). However, in most realistic cases (particularly in the case of images), the data looks far from being i.i.d.

For one-dimensional data, we model the dependence between adjacent samples by assuming that the observations and labels are generated by $(\mathcal{X}, \boldsymbol{\Theta}) = \{(X_i, \Theta_i)\}$, where $\boldsymbol{\Theta}$ is a stationary process generating the labels, and the observations X_i are conditionally independent given the corresponding labels θ_i, and distributed according to $f_{\theta_i}(\cdot)$, where $f_0(\cdot)$ and $f_1(\cdot)$ are fixed, known densities. In this setting, the problem of classification consists of observing the sequence of values $\{\mathbf{x}_i\}$ and producing a corresponding sequence of labels $\{\hat{\theta}_i\}$, in such a way as to minimize the *error rate*, $\lim_{n\to\infty} \mathrm{E}\left[n^{-1} \sum_{i=1}^{n} \delta_{\hat{\theta}_i, \theta_i}\right]$. Because of stationarity, the error rate is equal to the probability of classification error $\Pr\left(\hat{\theta}_i \neq \theta_i\right)$. A classifier that, given a collection of observations $\{x_i\}$, assigns labels $\{\hat{\theta}_i\}$ to them independently of one another is called a *product classifier.*

2.2 The Progressive Classifier

Assume that the data is in the form of an L-level DWT multiresolution pyramid.

Definition 1 *A* level-L progressive classifier *is a pair* $(B, \{\Pi^i\}_{i=1}^{L})$*, where:*

i) B, the baseline classifier*, is the product classifier characterized by the partition* $\Pi^0 = \{\Pi_0^0, \Pi_1^0\}$ *of the observation space, and operates on the full-resolution data;*

ii) $\{\Pi^\ell\}_{\ell=1}^L$ *is a collection of partitions of the observation space such that, for each* ℓ*,* Π^ℓ *consists of three (distinct) regions* $\Pi_{prog}^\ell, \Pi_0^\ell, \Pi_1^\ell$.

The progressive classifier starts the analysis of the data at the coarsest resolution level, level L. If a coefficient w_i^L belongs to the region Π_j^L then all the samples of the corresponding block at full resolution are labeled with the same label j. If the coefficient falls within region Π_{prog}^L the progressive classifier first inverts the DWT one level (to get the coefficients $\mathbf{w}^{L-1}$ from $\mathbf{w}^L$ and $\mathbf{v}^L$), and then analyzes the corresponding coefficients at level $(L-1)$ independently, using the partition Π^{L-1}. The same process is repeated iteratively.

To simplify the analysis, throughout Sections 2 and 3 we will assume that the DWT is taken based on the Haar basis, although, as we briefly discuss in Section 4.3, the progressive classifier just defined can be applied to any orthonormal or biorthogonal wavelet basis. Below we list some further assumptions that will be used in some of our results (all of which will subsequently be relaxed).

1. *Gaussian*: The distributions of samples of classes 0 and 1 are Gaussian with unit variance and means $+1$ and -1, respectively.

2. *Markov*: The labels are generated by a stationary symmetric Markov chain with transition probabilities $p_{i|j} = p$ if $i \neq j$, and $p_{i|i} = (1-p)$.

3. *Bayes*: The baseline classifier is the product Bayes' decision rule.

Under the Gaussian assumption, the joint distribution of (X_1, X_2) is a mixture of four bivariate Gaussians, with identity covariance matrix and means $(2\Theta_1 - 1, 2\Theta_2 - 1)$. In general (with or without the Gaussian assumption), let μ_{θ_1,θ_2} be the conditional measure on the (x_1, x_2)-plane induced by the distribution of (X_1, X_2), given the values of the corresponding labels $(\Theta_1 = \theta_1, \Theta_2 = \theta_2)$. Write $\pi(i,j) = \Pr\{\Theta_1 = i, \Theta_2 = j\}$ for the mixture coefficients, so that, under the Markov assumption, $\pi(1,1) = \pi(0,0) = (1-p)/2$ and $\pi(1,0) = \pi(0,1) = p/2$.

2.3 Results for the Level-1 Progressive Classifier

In this section we demonstrate that, under certain conditions, the level-1 progressive classifier is more accurate and computationally more efficient than the corresponding baseline classifier. All the proofs are given in Appendix A.

In the simplest case, when all three assumptions 1, 2 and 3 are satisfied, it follows from Equation (1) that the coefficients W_i^1 are distributed according to a mixture of Gaussians, with density

$$\frac{1}{2}\,(1-p)\,\phi(x+\sqrt{2})+\frac{1}{2}\,(1-p)\,\phi(x-\sqrt{2})+p\,\phi(x),$$

where $\phi(\cdot)$ is the standard Normal density. In this case the baseline classifier decides $\hat{\theta}_i = 1$ if $x_i > 0$ and $\hat{\theta}_i = 0$ otherwise. The level-1 progressive classifier operates as follows. The level-1 partition is defined by a threshold T, discussed more in detail in Section 4.2. If $w_i^1 < -T$ then the classifier decides $\hat{\theta}_{2i} = \hat{\theta}_{2i+1} = 0$, if $w_i^1 > T$ it decides $\hat{\theta}_{2i} = \hat{\theta}_{2i+1} = 1$, and if $-T \le w_i^1 \le T$ it takes the progressive step and reduces to the baseline classifier. Figure 1 shows the decision regions of the baseline classifier and the level-1 progressive classifier.

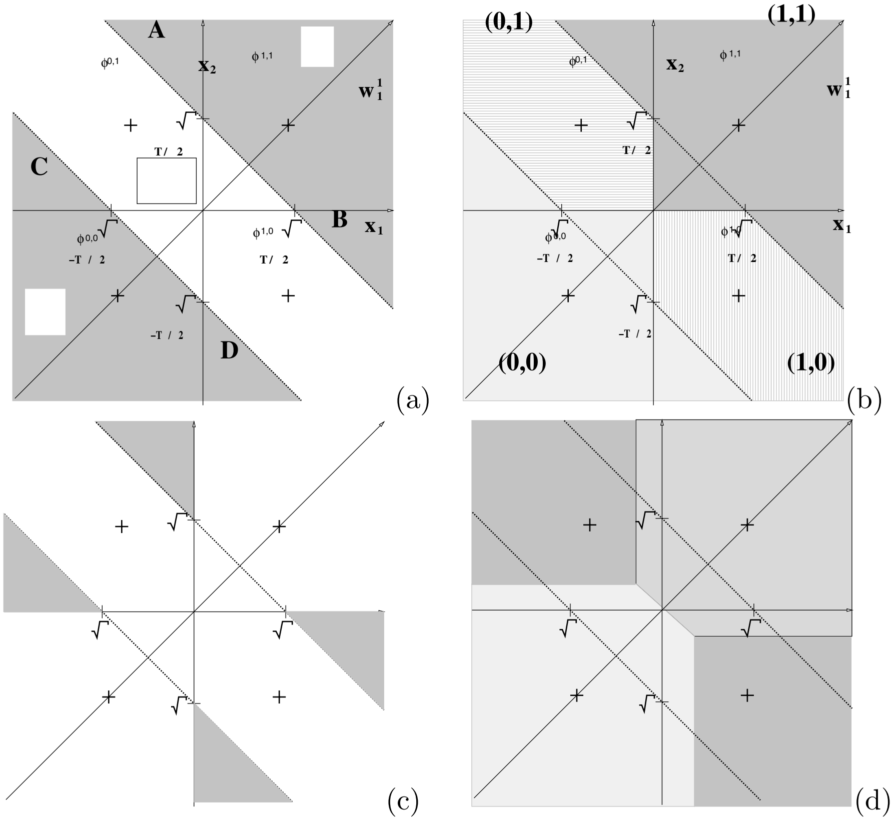

Figure 1:

Under these assumptions, the error rate of the level-1 progressive classifier with a fixed threshold is smaller than the error rate of the baseline classifier, for for data with strong enough correlations.

Theorem 1 *Under assumptions 1, 2, and 3, for each threshold T there exists $p_T > 0$ such that for every transition probability $p < p_T$ the error rate of the level-1 progressive classifier is smaller than the error rate of the baseline classifier.*

The theorem thus states that for every choice of the threshold T governing the progressive behavior, there is a family of label processes for which the level-1 progressive classifier is more accurate than the baseline classifier.

Next we drop the Gaussian and the Bayes assumptions (assumptions 1 and 3) and extend Theorem 1 to general level-1 progressive classifiers. Our next result makes it possible to compare baseline non-parametric classifiers (such as CART) with their corresponding progressive versions.

We partition the observation space (x_1, x_2) into disjoint regions $A_{i,j,k}$ and A_0, for $i, j, k \in \{0, 1\}$, where if the point (x_1, x_2) belongs to region $A_{i,j,k}$, the baseline classifier decides $\hat{\theta}_1 = i, \hat{\theta}_2 = j$ and the level-1 progressive classifier decides $\hat{\theta}_1 = \hat{\theta}_2 = k$. If the point (x_1, x_2) belongs to region A_0, the progressive classifier takes the progressive step, and the baseline classifier and the progressive classifier produce the same results. If a point falls into any region $A_{i,j,k}$ where $i \neq k$ or $j \neq k$, the progressive classifier and the baseline classifier produce different results. The following theorem states conditions under which the progressive classifier is more accurate than the baseline classifier in terms of the probabilities of such regions.

Theorem 2 *Under the Markov assumption 2, if*

$$P_1 = \sum_{m=0,1} \sum_{i,j=0,1} \left\{ (1 - 2\delta_{i,m})\mu_{m,m}(A_{i,j,\bar{i}}) + (1 - 2\delta_{j,m})\mu_{m,m}(A_{i,j,\bar{j}}) \right\} > 0, \tag{2}$$

then there exists $p_0 > 0$ such that for each transition probability $p < p_0$ the level-1 progressive classifier has smaller error rate than the baseline classifier.

Although the assumption $P_1 > 0$ may at first seem like an arbitrary technicality, a closer look shows that, in general, only an erroneously constructed progressive classifier will have $P_1 \leq 0$ (observe that we can interchange the roles of i and j in (2), i.e., change the decisions of the progressive classifier in the regions $A_{i,j,k},\ k! = i$ and $A_{i,j,k},\ k! = j$ to their complements, to obtain a classifier satisfying the condition.) Thus, any bona fide progressive classifier is more accurate than the baseline classifier, as long as the label process is correlated enough.

Theorem 2 is especially useful in the construction of non-parametric progressive classifiers, where the condition 2 can be easily checked, and confidence intervals for P_1 can be constructed using Monte Carlo methods. Also, once both the baseline and the progressive classifiers are trained, the value of p_0 can be estimated using iterative simulation techniques.

Next we obtain conditions under which the progressive classifier is faster than the baseline classifier.

Classification is an expensive operation. In practice, both parametric and nonparametric classifiers require considerably more resources than inverting a wavelet transform and, since classifying a sample at full resolution and classifying a wavelet coefficient have comparable computational complexity, the figure of merit will be the expected number of elementary classification operations.

Theorem 3 *The level-1 progressive classifier is faster than the corresponding baseline classifier if and only if $Pr\{W_1^1 \in \Pi_{prog}^1\} < 1/2$.*

For example, let assumptions 1, 2 and 3 hold, consider the Bayes progressive classifier based on a threshold T, let $\Phi(x)$ denote the standard normal cumulative distribution function, and define $Q_1 = \Phi(T) - \Phi(-T)$, and $Q_2 = \Phi(T - \sqrt{2}) - \Phi(-T - \sqrt{2}) < Q_1$. It is easy to see here that if $Q_1 < 1/2$, then the progressive classifier will be faster than the baseline classifier, and if $Q_2 > 1/2$ it will be slower, regardless of the value of the transition probability p. If, however, $Q_2 < 1/2$ and $Q_1 > 1/2$, then the progressive classifier will only be faster when $p < p_0 = (1/2 - Q_2)/(Q_1 - Q_2)$.

This example can be generalized as follows, by relaxing the Gaussian assumption to generic distributions:

Corollary 1 *Under the Markov assumption, suppose that the level-1 partition satisfies*

$$\mu_{1,0}(\Pi_{prog}^1) + \mu_{0,1}(\Pi_{prog}^1) > 1 \qquad \text{and} \qquad \mu_{1,1}(\Pi_{prog}^1) + \mu_{0,0}(\Pi_{prog}^1) < 1.$$

Then there exists a transition probability $p_0^{'} > 0$ such that for each $p < p_0^{'}$ the level-1 progressive classifier is faster than the baseline classifier.

The main results of this section, Theorem 2 and Corollary 1, imply that, for generic class-conditional distributions of the observations and generic classifiers, under just the Markov assumption, when the transition probability p of the label process is small enough, a well-designed level-1 progressive classifier is both faster and more accurate than the corresponding baseline classifier.

2.4 Results for the Level-ℓ Progressive Classifiers

Here we extend the results of the previous section to level-ℓ progressive classifiers. Combined with the results of the previous section they provide criteria for the selection of the starting level and the partitions of the progressive classifier (see also the remark at the end of Section 3.2). The proofs are given in Appendix A6.

From the definition of the Haar wavelet coefficients and the conditional independence assumption, it follows that the density of a level-ℓ coefficient W_i^ℓ is a scaled mixture of several copies of the densities f_0 and f_1. Given the values of the corresponding labels, the conditional density of W_i^ℓ only depends on $S_i^\ell = \sum_{j=2^\ell i}^{2^\ell (i+1)-1} \Theta_j$, the number of class-1 labels in the set $\{(X_{2^\ell i}, \Theta_{2^\ell i}), \ldots, (X_{2^\ell (i+1)-1}, \Theta_{2^\ell (i+1)-1})\}$. In particular, under the Gaussian assumption, the conditional density of $W_0^{\ell-1}$ given $\theta_0, \ldots, \theta_{2^{\ell-1}-1}$ is

$$\begin{aligned} f_{W_0^{\ell-1}}(w \mid \theta_0, \ldots, \theta_{2^{\ell-1}-1}) &= f_{W_0^{\ell-1}}(w \mid S_0^{\ell-1}) & (3) \\ &= \phi\left(w + 2^{-(\ell-1)/2}(2^{\ell-1} - 2S_0^{\ell-1})\right), & (4) \end{aligned}$$

and its unconditional density is given by the mixture

$$f_{W_0^{\ell-1}}(w) = \sum_{k=0}^{2^\ell-1} f_{W_0^{\ell-1}}(w \mid S_0^{\ell-1} = k)\mathrm{Pr}\left\{S_0^{\ell-1} = k\right\}, \tag{5}$$

where the mixing coefficients $\mathrm{Pr}\{S_0^{\ell-1} = k\}$ can be calculated from the model for the label process $\boldsymbol{\Theta}$. Incidentally, note that equations (3) and (5) still define the densities of the wavelet coefficients when the Gaussianity assumption 1 is relaxed. In particular, $f_{W_0^{\ell-1}}(w \mid \theta_0, \ldots, \theta_{2^{\ell-1}-1})$ depends only on the number of samples of class 1, that is, on $S_0^{\ell-1}$. However, it is clear that $f_{W_0^{\ell-1}}(w \mid S_0^{\ell-1} = k)$ is no longer a Gaussian density, and therefore equality (4) does not hold.

Similarly, the joint density $f_{W_0^{\ell-1},W_1^{\ell-1}}(w_0, w_1)$ of $W_0^{\ell-1}$ and $W_1^{\ell-1}$ is given by

$$\sum_{i=0}^{2^\ell-1}\sum_{j=0}^{2^\ell-1} \mathrm{Pr}\{S_0^{\ell-1} = i, S_1^{\ell-1} = j\} f_{W_0^{\ell-1}}(w_0 \mid S_0^{\ell-1} = i) f_{W_1^{\ell-1}}(w_1 \mid S_1^{\ell-1} = j). \tag{6}$$

Although in general it is difficult to obtain closed-form expressions for the probabilities of the events $\{S_0^{\ell-1} = i, S_1^{\ell-1} = j\}$, in the Markov case standard numerical techniques can be used to easily provide adequate approximations (details can be found in Castelli and Kontoyiannis [4]). Figure 2 shows an example of the mixture density (6) and its components for level $\ell = 4$ under the Gaussian assumption, for the Markov model with $p = 0.08$.

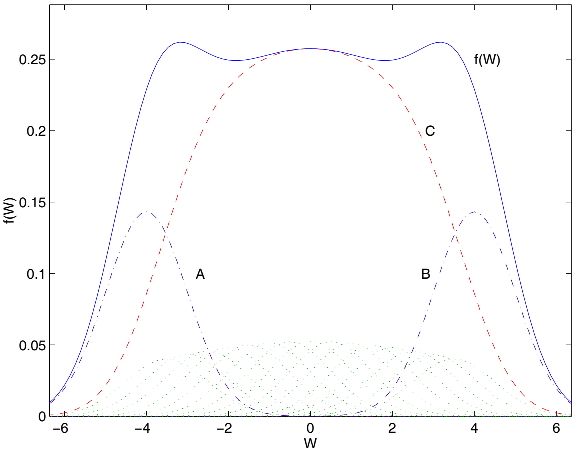

Figure 2:

The level-ℓ progressive classifier is completely specified by a level-$(\ell-1)$ progressive classifier and a partition of the (w_0, w_1)-plane into Π_0^ℓ, Π_1^ℓ and Π_{prog}^ℓ. If $W_0^\ell \in \Pi_{\mathrm{prog}}$, then the level-$\ell$ progressive classifier takes the progressive step and reduces to the product level-$(\ell-1)$ progressive classifier. To facilitate our analysis, we further subdivide Π_0^ℓ and Π_1^ℓ into the regions $\{A_{i,j,k}\}$, $k = 0, 1$, $i, j \in \{0, 1, \mathrm{prog}\}$. The index k indicates the decision of the level-ℓ classifier to label all the points in the corresponding full-resolution block as class k. The indices i and j denote the decisions of the level-$(\ell-1)$ classifier applied to $W_0^{\ell-1}$ and $W_1^{\ell-1}$, respectively. This means that if i equals 0 or 1, the classifier labels the block corresponding to $W_0^{\ell-1}$ as homogeneous of class i, and if $i = \text{prog}$ the classifier takes the progressive step; similarly for j.

Next we prove results analogous to Theorems 2 and 3. Let $\pi^\ell(i,j)$ denote the mixture parameters $\Pr\{S_0^{\ell-1} = i, S_1^{\ell-1} = j\}$, let $\pi^\ell(i) = \Pr\{S_0^\ell = i\}$, and write $\mu_{i,j}$ for the conditional measure on the (w_0, w_1)-plane induced by the density

$$f_{W_0^{\ell-1}, W_1^{\ell-1}}(w_0, w_1 \mid S_0^{\ell-1} = i, S_1^{\ell-1} = j) \;=\; f_{W_0^{\ell-1}}(w_0 \mid S_0^{\ell-1} = i) f_{W_1^{\ell-1}}(w_1 \mid S_1^{\ell-1} = j).$$

Let $e_{i,j,k}$ denote the conditional expected number of errors that the level-$(\ell-1)$ classifier makes in classifying $W_0^{\ell-1}$ and $W_1^{\ell-1}$, given that $W_0^\ell \in A_{i,j,k}$.

Theorem 4 *The error rate of the level-ℓ progressive classifier is smaller than the error rate of the level-$(\ell-1)$ progressive classifier if and only if the following quantity is positive:*

$$\sum_{i,j} \sum_{n=0}^{2^\ell} \sum_{m=\max(0,2^\ell-n)}^{\min(n,2^\ell)} \pi(m, n-m) \Big\{ \mu_{m,n-m}(A_{i,j,1})(e_{i,j,1} + m - 2^\ell) + \mu_{m,n-m}(A_{i,j,0})(e_{i,j,0} - m) \Big\}.$$

Theorem 5 *Any level-ℓ progressive classifier that has $Pr\{W_1^\ell \in \Pi_{prog}^\ell\} < 1/2$, is faster than the corresponding level-$(\ell-1)$ progressive classifier.*

In the Gaussian case with the Bayes classifier and a Markov model, Theorem 5 simplifies to:

Corollary 2 *Suppose that assumptions 1, 2 and 3. There exists $p_\ell > 0$ such that for each $p < p_\ell$ the level-ℓ progressive classifier is faster than the corresponding level-$(\ell-1)$ progressive classifier.*

3 Classification in Two Dimensions

3.1 Preliminaries

For two-dimensional data ("images"), we consider the DWT performed using separable filters; the image is first filtered and downsampled row-by-row and the resulting matrix is filtered and downsampled column-by-column. The result is arranged as a matrix composed of four blocks (subbands). The upper-left block, called the *lowest frequency subband at level 1*, contains an approximation of the original image at half resolution and double scale. The filtering operation is performed recursively on this subband.

In this section we carry over the analysis of Section 2, based on the more realistic image model of a dependent random field. Consider a collection of labeled samples generated by the stationary random field $(\mathcal{X}, \mathbf{\Theta}) = \{(\mathbf{X}_u, \Theta_u) \ ; \ u \in \mathbf{Z}^2\}$, where the observations X_i take values in $\mathbf{R}^d$ and the labels $\Theta_u \in \{0,1\}$. Assume that, given the values of the corresponding class labels, the random variables $\mathbf{X}_u$ are conditionally independent, and distributed according to the fixed densities $f_0(\cdot)$ and $f_1(\cdot)$. The general level-ℓ progressive classifier is defined exactly as in the one-dimensional case. The other assumptions 1 and 3 from Section 2.2 remain the same, while the Markovian assumption is replaced by the stationarity of the label field.

As before, an important quantity in the analysis will be S_u^ℓ, the number of class-1 samples corresponding to W_u^ℓ. For $u = (m,n)$, $S_u^\ell = S_{m,n}^\ell = \sum_{i=2^\ell m}^{(2^\ell+1)m-1} \sum_{j=2^\ell n}^{(2^\ell+1)n-1} \Theta_{i,j}$. By the definition of the Haar basis, (cf. (1) in one dimension),

$$\begin{aligned} W_{m,n}^\ell &= \frac{1}{2} W_{2m,2n}^{\ell-1} + \frac{1}{2} W_{2m+1,2m}^{\ell-1} + \frac{1}{2} W_{2m,2n+1}^{\ell-1} + \frac{1}{2} W_{2m+1,2n+1}^{\ell-1} \\ &= 2^\ell \frac{1}{2^{2\ell}} \sum_{i=0}^{2^\ell-1} \sum_{j=0}^{2^\ell-1} X_{2^\ell m+i, 2^\ell n+j} \end{aligned}$$

so that the conditional density of $W_{m,n}^\ell$, given the labels $\{\theta_{i,j} \ ; \ i = 2^\ell m, 2^\ell m+1, \ldots, 2^\ell(m+1)-1, \ j = 2^\ell n, \ldots, 2^\ell(n+1)-1\}$, is a scaled convolution of $S_{m,n}^\ell$ copies of f_1 and $(2^{2\ell} - S_{m,n}^\ell)$ copies of f_0.

3.2 Results

In this section we generalize Theorems 4 and 5 of Section 2.4 to the general level-ℓ progressive classifiers operating on two-dimensional data. We obtain conditions under which the progressive level-ℓ classifier is more accurate and more efficient than the corresponding level-$(\ell-1)$ classifier. The proofs of Theorems 6 and 7 below are exactly parallel to those of Theorems 4 and 5, and are omitted.

Let $f_{i,j,h,k}^\ell(w_{0,0}, w_{0,1}, w_{1,0}, w_{1,1})$ be the joint conditional density of $(W_{0,0}^{\ell-1}, W_{0,1}^{\ell-1}, W_{1,0}^{\ell-1}, W_{1,1}^{\ell-1})$ given $\{S_{0,0}^{\ell-1} = i, S_{0,1}^{\ell-1} = j, S_{1,0}^{\ell-1} = h, S_{1,1}^{\ell-1} = k\}$, and denote the induced conditional measure by $\mu_{i,j,h,k}^\ell(w_{0,0}, w_{0,1}, w_{1,0}, w_{1,1})$. Similarly, $f_i^\ell(w)$ denotes the conditional density of W_u^ℓ given $\{S_u^\ell = i\}$, and $\mu_i^\ell(w)$ is the induced measure. As before, the mixture coefficients are written as $\pi^\ell(i) = \Pr\{S_u^\ell = i\}$ and $\pi^\ell(i,j,h,k) = \Pr\{S_{0,0}^{\ell-1} = i, S_{0,1}^{\ell-1} = j, S_{1,0}^{\ell-1} = h, S_{1,1}^{\ell-1} = k\}$.

Now fix a level-$(\ell-1)$ classifier with partition $\Pi_0^{\ell-1}, \Pi_1^{\ell-1}, \Pi_{\text{prog}}^{\ell-1}$, and consider a level-$\ell$ partition $\Pi_0^\ell, \Pi_1^\ell, \Pi_{\text{prog}}^\ell$ for the level-ℓ classifier. Define the regions:

$$A_{i,j,h,k,m} = \{w_{0,0}^{\ell-1} \in \Pi_i^{\ell-1}, w_{0,1}^{\ell-1} \in \Pi_j^{\ell-1}, w_{1,0}^{\ell-1} \in \Pi_h^{\ell-1}, w_{1,1}^{\ell-1} \in \Pi_k^{\ell-1}, w^\ell \in \Pi_m^{\ell-1}\},$$

and write $e_{i,j,h,k,m}$ for the conditional expected number of errors of the product level-$(\ell-1)$ classifier given that $(w_{0,0}^{\ell-1}, w_{0,1}^{\ell-1}, w_{1,0}^{\ell-1}, w_{1,1}^{\ell-1}) \in A_{i,j,h,k,m}$.

Our next result extends Theorem 4 to the two-dimensional case.

Theorem 6 *If the partition $\Pi_0^\ell, \Pi_1^\ell, \Pi_{prog}^\ell$ of the level-ℓ classifier satisfies*

$$\sum_{i,j,h,k} \sum_{n=1}^{4^\ell - 1} \sum_{\substack{\{n_1,\, n_2,\, n_3,\, n_4\} \\ \Sigma n_i \,=\, n}} \pi^\ell(n_1, n_2, n_3, n_4) \Big[\mu_{n_1,n_2,n_3,n_4}\left(A_{i,j,h,k,1}\right) \Big\{ e_{i,j,h,k,1} + \sum n_i - 4^\ell \Big\}$$

$$+\mu_{n_1,n_2,n_3,n_4}\left(A_{i,j,h,k,0}\right) \Big\{ e_{i,j,h,k,0} - \sum n_i \Big\} \Big] \quad > \quad 0,$$

then the error rate of the level-ℓ progressive classifier is smaller than the error rate of the level-$(\ell - 1)$ classifier.

The above condition is not as clean as in the one-dimensional case, partly because no assumptions were made on the label field $\mathbf{\Theta}$ so we do not have exact expressions for the mixture parameters $\pi^\ell(i, j, h, k)$. However, all the listed quantities can be consistently estimated from the data, using simple estimators, making the result of Theorem 5 especially valuable for non-parametric classifiers. Consider, for example, the following procedure: Divide the available samples into a training set and test set. Train the classifier with the training set, then restrict the attention to the level-ℓ wavelet coefficients w^ℓ of the test set, that are generated from four level-$(\ell - 1)$ coefficients ($w_{0,0}^{\ell-1}$, $w_{0,1}^{\ell-1}$, $w_{1,0}^{\ell-1}$, and $w_{1,1}^{\ell-1}$) corresponding to regions of the image containing n_1, n_2, n_3 and n_4 samples of class 1 respectively. Compute the proportion of such coefficients for which $w_{0,0}^{\ell-1} \in \Pi_i^{\ell-1}$, $w_{0,1}^{\ell-1} \in \Pi_j^{\ell-1}$, $w_{1,0}^{\ell-1} \in \Pi_h^{\ell-1}$, $w_{1,1}^{\ell-1} \in \Pi_k^{\ell-1}$, and $w^\ell \in \Pi_1^\ell$. This proportion is a consistent estimator of $\mu_{n_1,n_2,n_3,n_4}\left(A_{i,j,h,k,1}\right)$.

When a parametric model is assumed, the above quantities can be estimated numerically, and confidence intervals found, for instance through Monte Carlo simulation. For example, in Castelli and Kontoyiannis [4] a simple, explicit construction of an m-dependent field $\mathbf{\Theta}$ was used, which was motivated by the problem of classifying satellite images of the earth's surface. This construction provided us with a field that looked very similar (in the statistical sense) to the distribution of land-cover classes found in practice, and also allowed for direct simulation, and easy calculation, of the prior probabilities $\pi^\ell(n)$ and $\pi^\ell(i, j, h, k)$ needed for the construction of parametric classifiers.

Note that the result of Theorem 6 can be easily specialized to any set of additional assumptions. Under the Gaussian assumption, for instance, if the progressive classifier is defined by a threshold (as it is the case for the Bayes decision rule) we immediately obtain the following.

Corollary 3 *Suppose that the Gaussian assumption holds, that the baseline classifier is defined by a threshold T, and that there exists a value of T, say $T^* < \infty$, such that the level-ℓ progressive classifier has smaller error rate than the level-$(\ell - 1)$ classifier.*

Then the same will hold for any $T > T^*$, and there exists a unique threshold $T^{opt} < \infty$ at which the difference of the error rate is maximum.

The analogue of Theorem 5 in the two-dimensional case is:

Theorem 7 *Any level-ℓ progressive classifier satisfying $Pr\{W_u^\ell \in \Pi_{prog}^\ell\} < 3/4$ is faster than the corresponding level-$(\ell - 1)$ progressive classifier.*

A remark on the selection of the starting level: Since the random variable $2^{-\ell}(S_{\mathbf{0}}^\ell - 2^{2(\ell-1)})$ converges to a Gaussian with finite, nonzero variance as $\ell \to \infty$ (see, for example, Guyon [13] for a general central limit theorem), the probabilities of the events $\{S_{\mathbf{0}} = 0\}$ and $\{S_{\mathbf{0}} = 2^{2\ell}\}$ both tend to zero. Therefore, starting progressive classification at a very high level ℓ of the multiresolution pyramid, will almost always force us to take the progressive step.

4 Extensions

4.1 Speed vs. Accuracy

In certain applications, minimizing the probability of classification error alone may not be the most satisfactory approach. Also from our results it becomes apparent that accuracy and speed are contrasting requirements: Reducing the size of the progressive regions will make the progressive classifier faster, but this eventually reduces the accuracy. In order to balance the tradeoff between classification accuracy and classification time we can introduce a Lagrangian cost function, $J(\lambda) = R + \lambda t$, where R is the probability that a sample is incorrectly classified, and t is the average classification time. Recall that t is a function of the starting level in the multiresolution pyramid, of the hierarchy of decision regions, and of the implementation of the DWT algorithm, and R is a function of the starting level and the decision regions. In practice, this loss function is easy to implement and tune: Large values of λ will produce strategies that most of the time classify at the lowest available level of the pyramid, while small values of λ will give classifiers that take the progressive step more often. Figure 3 shows an example of the form of J, for a 1-level progressive classifier, under the assumptions of Theorem 1. Here, the progressive classifier is uniquely determined by a threshold T, thus J depends only on T and λ. Note that, for every choice of λ there is a T_λ that minimizes J, which corresponds to the unique solution to the minimization problem.

When implementing progressive classifiers, the size of the progressive region can be easily tuned. In the nonparametric case this is accomplished by varying the percentage of training samples corresponding to regions of the training data where samples from both classes are present. In the parametric

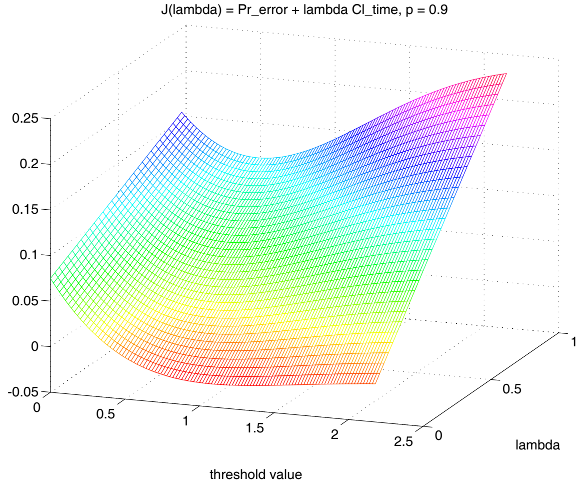

Figure 3:

case the baseline classifier produces estimates of the posterior probabilities, so that a final decision is always reached by means of a likelihood ratio test. Thus the amount of progressiveness can be tuned by selecting appropriate thresholds for the ratio test.

4.2 Higher-Dimensional Data

The progressive classifier, as well as the present analysis can be extended to higher-dimensional data, as long as they are naturally indexed by (a rectangular subset of) the $d-$dimensional Euclidean lattice. Practical examples are data from volumetric (three-dimensional) medical imaging, geological data or atmospheric data. Also we have only considered scalar predictors, although vector-valued predictors are also very common. In the case of multispectral satellite images, each pixel location is associated with a vector of measurements corresponding to the intensity of the electromagnetic radiation in different portions of the spectrum. Since the derivations of Theorems 2 through 7 do not rely on the scalar predictors assumption, the results hold in the more general vector-valued case.

4.3 General Wavelet Bases

Although the Haar wavelet basis was used to simplify the theoretical analysis, in practice, progressive classification can be applied using any orthonormal or biorthogonal family of wavelets, with only minor modifications. As reported in Castelli *et al.* [4] a large number of experiments were done using other wavelet bases, and the experimental results agree well with our theoretical findings.

In the general case the analysis will be more involved since adjacent wavelet coefficients are no longer conditionally independent given the values of the underlying labels. But as the proofs of the more general results (Theorems 2–5 and their two-dimensional counterparts) did not explicitly rely on that assumption, they remain valid in the generic wavelet case with only minor modifications in their statements.

The extension is based on the following argument: if we use a scaling function filter with $2n$ coefficients, a coefficient at level ℓ depends on the values of N_ℓ pixels. In one dimension, N^1_ℓ can be computed recursively, as follows: $N^1_1 = 2n$, $N^1_\ell = 2 * (2n + N^1_{\ell-1} - 2]$. In k dimensions, $N^k_\ell = (N^1_\ell)^k$, since we use separable filters. Thus, the conditional distribution of the coefficient given the labels of the corresponding hypercubic set of pixels depends now on N^k_ℓ labels. Divide the labels into those corresponding to the "core" cube of side 2^ℓ, and the remaining ones. Take the expectation of this conditional distribution with respect to the non-core labels. The resulting distribution is the conditional distribution of the wavelet coefficient given the central $2^{\ell\, k}$ labels. Note that these labels are now unique to the coefficient, i.e., the corresponding conditional distribution of the adjacent wavelet coefficients is conditional given a disjoint set of labels, as it was the case with the Haar wavelet. Substitute then this distribution for the one used in Theorems 2–5 (and their two-dimensional counterparts), modify accordingly the definitions of the measures μ. Then the formal results and their proofs hold without change.

Although they may not always provide computationally simple ways to define the optimum decision regions in the parametric case, Theorems 2–7 justify the use of the progressive approach in many relevant practical cases, most importantly in conjunction with nonparametric classifiers. In this case they can also provide guidelines on how to evaluate numerically the partitions produced by nonparametric methods.

5 An Extreme Case

While this paper is not meant as a case study, we include the results of a small experiment comparing the progressive classifier with the baseline classifier, and with a more sophisticated method called ICM (Iterated Conditional Modes) described in [2]. The experiment was designed on purpose in such a way as to favor ICM, so this case is really a stretch for the progressive classifier.

ICM relies on the same assumptions as the progressive classifier, namely, that the label field is a stationary stochastic process and that the observations are conditionally independent given the corresponding labels. ICM is based on a classifier that produces estimates of the posterior probabilities of the labels given the observations, and of the joint distribution of the labels in the immediate neighborhood of each pixel. In a first pass, the posteriors of the classes given the observations are computed, and then the resulting map is iteratively refined using the estimates of the local joint distributions of neighboring pixels. This method was suggested to us because it converges quickly and produces apparently good results.

The data is a portion of a satellite image from the NALC dataset, of size 1000×1000 pixels, where each pixel covers 60×60 meters square on the ground. This image covers a portion of the Colorado high planes, and was acquired during the month of August.

To favor ICM, we selected a two-class classification problem; in an n-class problem, even though ICM uses a simple 3×3 neighborhood region, our implementation of ICM would require the reliable estimation of n^8 conditional probabilities. The two classes were selected in such a way that the baseline classifier could distinguish them well. In particular, we selected agricultural regions and non-agricultural regions. Vegetated areas other that non-agricultural regions appear less "green" (lighter) in the image (see Figure 4).

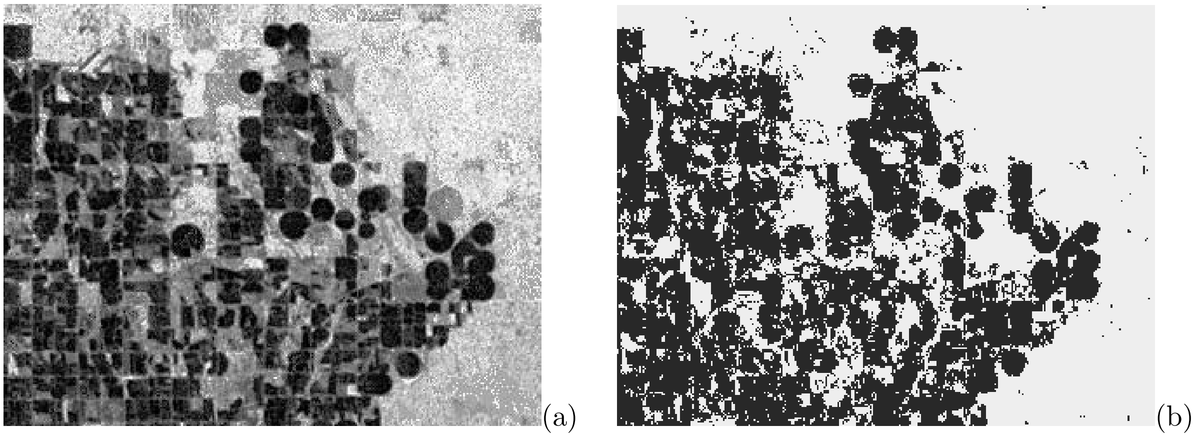

Figure 4:

The test image was then manually classified using computer-assisted classical photo-interpretation methods, and a ground-truth map was produced. The conditional distributions used by ICM were computed from the entire 1000 × 1000 pixel image. In the experiments the same baseline classifier is used alone and in conjunction with ICM. Training sets were constructed by randomly sampling selected portions of the 1000 × 1000 image. Samples selected for the training set were not used to evaluate the accuracy of the classification.

A training set size of 8000 samples was used for the baseline classifier, for the ICM classifier, and for the overall progressive classifier. In the case of the progressive classifier the 8000 samples were equally divided between the different levels. We also tried using 8000 samples for each level; this did not significantly slow down the progressive classifier, although, of course, it increased the classification accuracy.

The baseline classifier used in the experiment was the 7-nearest-neighbor classifier [7], our implementation of which automatically produces estimates of the a-posteriori probabilities of the individual classes, and thus is well-suited for use with ICM. It is worth noting, that, unlike ICM, progressive classification can be used with classifiers that do not produce estimates of the posterior probabilities (such as CART [3]). In our experience, such classifiers are much faster and equally accurate.

Finally, our implementation of ICM relies on fast algorithms that we have developed in the past for similar purposes (segmentation rather than classification), thus the timing for ICM does not suffer from poor coding.

The multiresolution pyramid was obtained with the Daubechies Biorthogonal Symmetric wavelets of order 3, corresponding to filters of length 8. The experiments were run on an IBM workstation model 43P, with a PowerPC 604 processor running at 133 MHz, 192 MB of RAM and 1 MB of cache. Timings were obtained directly from the system clock, using appropriate UNIX system calls, and reflect only the time spent in the process address space.

Results are shown in Table 1. Given the medium resolution of the image data and the particular characteristics of the selected region, as the starting resolution decreases, the progressive classifier tends to take the progressive step rather than classifying the samples. Straightforward estimation of the quantities in Theorems 6 and 7 showed that the level-2 progressive classifier would be faster but less accurate than the level-1 progressive classifier, and that the level-1 progressive classifier would be both faster and more accurate than the baseline classifier. We do not know how to compare ICM with the progressive or baseline classifier other than through experiments.

From Table 1 it is clear that the straightforward (counting) estimates in Theorems 6 and 7 accurately predicted both speed and accuracy of the level 1 and level 2 progressive classifiers. The level-1 progressive classifier is both faster and more accurate than the baseline classifier. The level-2 classifier is faster than the level-1 classifier, yet slightly less accurate than the baseline classifier. ICM is by nature slower than the baseline classifier, more accurate (not surprisingly so, since we used a very large training set to estimate the random field local properties), and yet, it appears not to offer any advantage over the progressive classifier.

The results of the first three columns were obtained using the 7-nearest neighbor classifier. The last column of Table 1 shows the speedup when a faster classifier is used, namely our implementation of CART [3], for which we have only a classification and not a regression part (thus, we could not use ICM in conjunction with CART. We assumed, however, that the post-processing time would be essentially the same as with the 7-nearest-neighbor classifier, given that the CART and 7-nearest-neighbor baseline classifier basically produce the same results. Here, the inherent speed of the baseline classifier somewhat reduces the speedup of the level-1 progressive classifier, since the progressive classification overhead costs are now of the same order of magnitude as the straight classification costs. However, the level-1 progressive classifier is now about 5 times faster than ICM, and the level-2 progressive classifier is more than 10 times faster.

It is clear that more complex classifiers will produce better results than our implementation of ICM in terms of accuracy. However, even in the extreme case studied here, the progressive classifier was not outperformed in terms of accuracy by a more complex algorithm. For the class of problems in which we are interested (mostly centered around automatic analysis of satellite images) where the amount data produced by the instrumentation is enormous and growing at an exponential rate, progressive classification is an appealing solution in terms of accuracy, even when compared to more elaborate classifiers than the ones almost universally used in the field.

6 Conclusions

We presented and analyzed *progressive classification*, a new classification framework for data organized on regular lattices, such as images. Progressive classification can be used in conjunction with essentially any existing classification rule. Since the progressive classifier was originally developed to label the pixels of large multispectral satellite images, our analysis is based on comparisons with the usual approach consisting of labeling individual pixels independently based on their reflectance in different spectral bands.

We stated and proved results showing that, under a wide range of assumptions, the progressive classifier is both more accurate and computationally more economical than corresponding baseline classifiers. Specifically, we showed that if the label process is sufficiently correlated, then the progressive classifier is both faster and more accurate than the simple pixel-by-pixel classification approach. Although the statements of the more general theorems seem rather complicated, all the quantities involved can be easily and consistently estimated from the data. This allows the development of simple algorithms for selecting the optimum number of levels of the multiresolution pyramid, given specific constraints on accuracy and speed.

The progressive classifier is not the optimum classifier; yet it appears to perform very well, even in comparison to more complex algorithms. In Section 5 we presented the results of an experimental comparison between the progressive classifier and a Bayesian classification method, ICM. The setup for the comparison was specifically designed in a way that should favor ICM. Our results clearly indicate that the progressive classifier was not outperformed in terms of accuracy, while it was significantly faster that ICM.

The combination of these theoretical and experimental findings suggest that the progressive classifier is an appealing solution for problems involving large amounts of image data, or when individual image sizes make it computationally infeasible to use (Bayesian) random field methods. Examples of such types of data include data used in the oil industry, remotely sensed images of the earth's surface, and aerial photographs.

Acknowledgements

We would like to thank John Turek and Chung-Sheng Li for several interesting discussions, and David Donoho and Andrew Barron for their insightful comments.
This work was funded in part by grant no. NASA/CAN NCC5-101.

Appendix A

A1 Proof of Theorem 1

Fix a threshold T and let $R_{\text{prog}} = R_{\text{prog}}(T,p)$ denote the error rate of the progressive classifier; similarly, write $R_{\text{bas}} = R_{\text{bas}}(T,p)$ for the error rate of the baseline classifier. Let $\Delta R = R_{\text{bas}} - R_{\text{prog}}$ and divide the half-space $\{(x_1 + x_2) > T\sqrt{2}\}$ into three regions, as shown in Figure 1(a): $A = \{(x_1+x_2) > T\sqrt{2}\} \bigcap \{x_1 \leq 0\}$, $B = \{(x_1+x_2) > T\sqrt{2}\} \bigcap \{x_2 \leq 0\}$ and $C = \{(x_1+x_2) > T\sqrt{2}\} \bigcap \{x_1 > 0\} \bigcap \{x_2 > 0\}$. Similarly define regions D, E and F in the half-space $\{(x_1 + x_2) < -T\sqrt{2}\}$.

When the point (x_1, x_2) belongs either to C or F then the level-1 progressive classifier reduces to the baseline classifier, so, in an obvious notation, $\Delta R = \Delta R(A) + \Delta R(B) + \Delta R(D) + \Delta R(E)$. From symmetry $\Delta R(A) = \Delta R(B) = \Delta R(D) = \Delta R(E)$, and so $\Delta R = 4\Delta R(A)$. When $(x_1, x_2) \in A$, the progressive classifier decides $\hat{\theta}_1 = \hat{\theta}_2 = 1$, while the baseline classifier decides $\hat{\theta}_1 = 1$ and $\hat{\theta}_2 = 0$. Since the error rate is the expected number of errors per decision,

$$\begin{aligned} R_{\text{prog}} &= 2\{0 \cdot \pi(1,1)\mu_{1,1}(A) + 1 \cdot \pi(1,0)\mu_{1,0}(A) + 1 \cdot \pi(0,1)\mu_{0,1}(A) + 2 \cdot \pi(0,0)\mu_{0,0}(A)\} \\ R_{\text{bas}} &= 2\{1 \cdot \pi(1,1)\mu_{1,1}(A) + 2 \cdot \pi(1,0)\mu_{1,0}(A) + 0 \cdot \pi(0,1)\mu_{0,1}(A) + 1 \cdot \pi(0,0)\mu_{0,0}(A)\}, \end{aligned}$$

so that, substituting for the mixture coefficients $\pi(i,j)$ and subtracting yields,

$$\Delta R = (1-p)(\mu_{1,1}(A) - \mu_{0,0}(A)) - p(\mu_{0,1}(A) - \mu_{1,0}(A)).$$

From the assumptions on the distributions of the samples it follows that both quantities $(\mu_{1,1}(A) - \mu_{0,0}(A))$ and $(\mu_{0,1}(A) - \mu_{1,0}(A))$ are positive. Consequently, for any p smaller than

$$p_T = \frac{\mu_{1,1}(A) - \mu_{0,0}(A)}{\mu_{1,1}(A) - \mu_{0,0}(A) + \mu_{0,1}(A) - \mu_{1,0}(A)},$$

the error rate of the progressive classifier is less than the error rate of the baseline classifier. □

A2 Proof of Theorem 2

When the level-1 progressive classifier takes the progressive step it reduces to the baseline classifier, and so their corresponding error rates can be written as

$$\begin{aligned} R_{\text{bas}} &= \frac{1}{2}\sum_{i,j}\sum_{m,n}\sum_{k}\left(2-\delta_{i,m}-\delta_{j,n}\right)\pi(m,n)\mu_{m,n}(A_{i,j})+R(A_0) \\ R_{\text{prog}} &= \frac{1}{2}\sum_{i,j}\sum_{m,n}\sum_{k}\left(2-\delta_{k,m}-\delta_{k,n}\right)\pi(m,n)\mu_{m,n}(A_{i,j,k})+R(A_0), \end{aligned}$$

where $R(A_0)$ is the component of the risk corresponding to the progressive step, and thus is common to both progressive level-1 and product baseline classifiers.

Next observe that $\delta_{k,m}-\delta_{i,m}=0$ when $i=k$, that $\delta_{k,m}-\delta_{i,m}=1$ when $k=m$ and $i=\overline{m}$, and that $\delta_{k,m}-\delta_{i,m}=-1$ when $k=\overline{m}$ and $i=m$; similar relations hold for $\delta_{k,n}-\delta_{j,n}$. So substituting for the probabilities $\pi(i,j)$, the difference $\Delta R = R_{\text{bas}}-R_{\text{prog}}$ becomes

$$\begin{aligned} &\frac{1}{2}\sum_{m,n}\sum_{i,j}\sum_{k}(\delta_{k,m}+\delta_{k,n}-\delta_{i,m}-\delta_{j,n})\pi(m,n)\mu_{m,n}(A_{i,j,k}) \\ =\ &\frac{1}{2}\sum_{m,n}\pi(m,n)\left[\sum_{i}\left\{(1-2\delta_{i,m})\sum_{j}\mu_{m,n}(A_{i,j,\bar{i}})\right\}+\sum_{j}\left\{(1-2\delta_{j,n})\sum_{i}\mu_{m,n}(A_{i,j,\bar{j}})\right\}\right] \\ =\ &\frac{1-p}{4}\sum_{m}\left[\sum_{i}\left\{(1-2\delta_{i,m})\sum_{j}\mu_{m,m}(A_{i,j,\bar{i}})\right\}+\sum_{j}\left\{(1-2\delta_{j,m})\sum_{i}\mu_{m,m}(A_{i,j,\bar{j}})\right\}\right] \\ &-\frac{p}{4}\sum_{m}\left[\sum_{i}\left\{(2\delta_{i,m}-1)\sum_{j}\mu_{m,\overline{m}}(A_{i,j,\bar{i}})\right\}+\sum_{j}\left\{(2\delta_{j,\overline{m}}-1)\sum_{i}\mu_{m,\overline{m}}(A_{i,j,\bar{j}})\right\}\right] \\ \triangleq\ &\frac{1-p}{4}P_1-\frac{p}{4}P_2. \end{aligned}$$

Since, by assumption, $P_1>0$, for each $p<p_0=P_1/(P_1+P_2)$ the difference ΔR of the error rates will be strictly positive. □

A3 Proof of Theorem 3

If $W_1^1 \notin \Pi^1_{\text{prog}}$, then the level-1 progressive classifier classifies the coefficient and no additional operations are required; conditional on the same event, the baseline classifier requires two operations. Now if $W_1^1 \in \Pi^1_{\text{prog}}$, then the level-1 classifier performs $1+2=3$ classification operations, and the baseline classifier performs only two. Thus, the level-1 classifier is faster if and only if

$$(1-\Pr\{W_1^1\in\Pi^1_{\text{prog}}\}+3\Pr\{W_1^1\in\Pi^1_{\text{prog}}\})\ <\ 2(1-\Pr\{W_1^1\in\Pi^1_{\text{prog}}\}+2\Pr\{W_1^1\in\Pi^1_{\text{prog}}\}),$$

namely, if and only if $\Pr\{W_1^1 \in \Pi^1_{\text{prog}}\} < 1/2$, as claimed. □

A4 Proof of Corollary 1

The level-1 progressive classifier requires 1 operation to label a pair of samples when the wavelet coefficient falls within the decision regions Π_0^1 or Π_1^1, and 3 operations for region Π^1_{prog}. Hence the expected number of operations to classify a sample (at full resolution) is

$$\begin{aligned}\frac{1}{2}\Pr\left\{W_1^1 \in \Pi_1^1\right\} \quad &+ \quad \frac{1}{2}\Pr\left\{W_1^1 \in \Pi_0^1\right\} + \frac{3}{2}\Pr\left\{W_1^1 \in \Pi^1_{\text{prog}}\right\} \\ &= \quad \frac{1}{2} + \Pr\left\{W_1^1 \in \Pi^1_{\text{prog}}\right\} \quad = \quad \frac{1}{2} + \sum_{i,j}\pi(i,j)\mu_{i,j}\left(\Pi^1_{\text{prog}}\right) \\ &= \quad \frac{1}{2} + \frac{1-p}{2}\left\{\mu_{1,1}\left(\Pi^1_{\text{prog}}\right) + \mu_{0,0}\left(\Pi^1_{\text{prog}}\right)\right\} \\ &\quad\; + \frac{p}{2}\left\{\mu_{1,0}\left(\Pi^1_{\text{prog}}\right) + \mu_{0,1}\left(\Pi^1_{\text{prog}}\right)\right\}.\end{aligned}$$

The progressive classifier will be faster than the baseline classifier if the expected number of operations is less than 1, and, rearranging, this inequality is equivalent to $p < p_0^{'}$, where

$$p_0^{'} \;=\; \frac{1-\mu_{1,1}\left(\Pi^1_{\text{prog}}\right) - \mu_{0,0}\left(\Pi^1_{\text{prog}}\right)}{\mu_{1,0}\left(\Pi^1_{\text{prog}}\right) + \mu_{0,1}\left(\Pi^1_{\text{prog}}\right) - \mu_{1,1}\left(\Pi^1_{\text{prog}}\right) - \mu_{0,0}\left(\Pi^1_{\text{prog}}\right)} \;>\; 0. \;\; \square$$

A5 Proof of Theorem 4

The decisions of the level-ℓ progressive classifier and the level-$(\ell-1)$ progressive classifier differ on the region $A = \bigcup A_{i,j,k}$. If we write $R^\ell(A) = \Pr\left\{\hat{\theta}_i \neq \theta_i \mid A\right\}\Pr(A)$ for the component of the error of the level-ℓ classifier in region A, then it can be expanded as:

$$\begin{aligned}&\sum_{n=0}^{2^\ell}\Pr\left\{A, \hat{\theta}_i \neq \theta_i \mid S_0^\ell = n\right\}\Pr\left\{S_0^\ell = n\right\} \\ &= 2^{-\ell}\sum_{n=0}^{2^\ell}\sum_{m=\max(0,2^\ell-n)}^{\min(n,2^\ell)}\left[\pi(m,n-m)\sum_{i,j}\left\{(2^\ell-n)\mu_{m,n-m}(A_{i,j,1}) + n\mu_{m,n-m}(A_{i,j,0})\right\}\right].\end{aligned}$$

For the product level-$(\ell-1)$ classifier, the corresponding quantity $R^{(\ell-1)}_{\text{prod}}(A)$ is given by

$$2^{-\ell}\sum_{n=0}^{2^\ell}\sum_{m=\max(0,2^\ell-n)}^{\min(n,2^\ell)}\left[\pi(m,n-m)\sum_{i,j}\{e_{i,j,1}\mu_{m,n-m}(A_{i,j,1}) + e_{i,j,0}\mu_{m,n-m}(A_{i,j,0})\}\right],$$

so the difference $\Delta R = R^{(\ell-1)}_{\text{prod}} - R^\ell$ is

$$2^{-\ell}\sum_{n=0}^{2^\ell}\sum_{m=\max 0,2^\ell-n}^{\min n,2^\ell}\left[\sum_{i,j}\pi(m,n-m)\left\{\mu_{m,n-m}(A_{i,j,1})(e_{i,j,1}-2^\ell+n)\right.\right.$$

$$+ \mu_{m,n-m}(A_{i,j,0})(e_{i,j,0} - n) \Bigg\} \Bigg] ,$$

and the theorem follows. □

A6 Proof of Theorem 5

The proof is similar to the proof of Theorem 3. If $W_1^\ell \notin \Pi_{\text{prog}}^\ell$, the level-$\ell$ progressive classifier classifies the coefficient, and no additional operations are required; conditional on the same event, the product level-($\ell-1$) classifier requires $2+n_1$ operations (the product level-($\ell-1$) classifier might take progressive steps). Conditional on $W_1^\ell \in \Pi_{\text{prog}}^\ell$, the product level-($\ell - 1$) classifier performs $2 + n_2$ classification operations, and the level-ℓ classifier performs $1 + 2 + n_2$ operations. Thus, the level-ℓ classifier is faster if $(1 - \Pr\{\Pi_{\text{prog}}^\ell\}) + (3 + n_1)\Pr\{\Pi_{\text{prog}}^\ell\})$ is smaller than $(2 + n_2)(1 - \Pr\{\Pi_{\text{prog}}^\ell\}) + (2 + n_1)\Pr\{\Pi_{\text{prog}}^\ell\})$, or, equivalently, if

$$(2 + n_2)\Pr\{\Pi_{\text{prog}}^\ell\} < 1 + n_2.$$

Clearly this is satisfied if $\Pr\{\Pi_{\text{prog}}^\ell\} < 1/2$, and the level-$\ell$ classifier is faster, independently of the choice of the level-($\ell - 1$) classifier. □

A7 Proof of Corollary 2

We just give a sketch of the proof. Note that, for each i, $\mu_{2^\ell/2,2^\ell/2}(\Pi_{\text{prog}}) \geq \mu_{i,2^n-j}(\Pi_{\text{prog}})$, and also that for any $p > 0$, $\pi(2^\ell/2, 2^\ell/2) \leq 1 - (1 - p)^{\ell-1}$. Arguing as in the proof of Corollary 1, the level-ℓ classifier is faster than the product level-($\ell - 1$) classifier if

$$\frac{1}{2}(1 - p)^{\ell-1} \; \Big\{\mu_{0,2^\ell}(\Pi_{\text{prog}}) + \mu_{2^\ell,0}(\Pi_{\text{prog}})\Big\} + \Big\{1 - (1 - p)^{\ell-1}\Big\} \mu_{2^{\ell/2},2^{\ell/2}}(\Pi_{\text{prog}}) < 1/2,$$

and solving this inequality for p this is equivalent to $p < p_\ell$, where

$$p_\ell = 1 - \left[\frac{\mu_{2^{\ell-1},2^{\ell-1}}(\Pi_{\text{prog}}) - 1/2}{\mu_{2^{\ell-1},2^{\ell-1}}(\Pi_{\text{prog}}) - \frac{1}{2}\Big\{\mu_{2^\ell,0}(\Pi_{\text{prog}}) + \mu_{0,2^\ell}(\Pi_{\text{prog}})\Big\}} \right]^{1/(\ell-1)} . \quad \square$$

References

[1] A. Antoniadis and G. Oppenheim (editors). *Wavelets and Statistics*. Lecture Notes in Statistics. Springer-Verlag, New York, 1995.

[2] J. Besag. On the statistical analysis of dirty pictures. *J. R. Statist. Soc. B*, 48(3):259–302, 1986.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth & Brooks/Cole, 1984.

[4] V. Castelli, I. Kontoyiannis, C.-S. Li, and J. J. Turek. Progressive classification: A multiresolution approach. Research Report RC 20475, IBM, 06/10/1996.

[5] S. R. Chettri and R. F. Cromp. Probabilistic neural network architecture for high-speed classification of remotely sensed imagery. *Telematics and Informatics*, 10(3), 1993.

[6] S. R. Chettri, R. F. Cromp, and M. Birmingham. Design of neural networks for classification of remotely sensed imagery. *Telematics and Informatics*, 9(3/4):145–156, Summer/Fall 1992.

[7] V. Dasarathy, Belur, editor. *Nearest Neighbor Pattern Classification Techniques.* IEEE Computer Society, 1991.

[8] I. Daubechies. *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[9] I. Daubechies, S. Mallat, and A.S. Willsky (editors). *Special Issue on Wavelet Transforms, IEEE Trans. Inform.Theory*, volume 38, nr. 2, part II. 1992.

[10] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition.* Springer, New York, 1996.

[11] D. L. Donoho. Cart and best-ortho-basis: a connection. *Ann. Stat.*, 25:1870–1911, 1997.

[12] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[13] Xavier Guyon. *Random Fields on a Network. Modeling, Statistics, and Applications.* Springer Verlag, 1995.

[14] L. D. Bergman et al. PetroSPIRE: A multi-modal content-based retrieval system for petroleum applications. In *3846,* Multimedia Storage Arch. Sys , 1999.

[15] L. D. Bergman et al. SPIRE, a digital library for scientific information. *Special Issue of IJODL, "in the tradition of Alexandrian Scholars"*, 1999. To appear.

[16] Justin D. Paola and Robert A. Schowengerdt. A detailed comparison of backpropagation neural network and maximum-likelihood classifiers for urban land-use classification. *IEEE Transaction on Geoscience and Remote Sensing*, 33(4):981–998, 1995.

[17] John A. Richards. *Remote Sensing Digital Image Analysis, an Introduction.* Springer-Verlag, 2nd edition, 1993.

[18] O. Rioul. Discrete-time multiresolution theory. *IEEE Trans. Sig. Proc.*, 41(8):2591–2606, August 1993.

[19] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE Signal Process Magazine*, 8(4):14–38, October 1991.

[20] Yehuda Salu and James Tilton. Classification of multispectral image data by the binary diamond neural network and by nonparametric, pixel-by-pixel methods. *IEEE Transactions on Geoscience and Remote Sensing*, 31(3):606–616, May 1993.

[21] R. Tate, D. Watson, and S. Eglen. Using wavelets for classifying human in vivo magnetic resonance spectra. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, Lecture Notes in Statistics, pages 377–383. Springer-Verlag, New York, 1995.

[22] V. Castelli et al. Progressive search and retrieval in large image archives. *IBM Journal of Research and Development*, 42(2):253–268, March 1998.

[23] V. Castelli et al. Search and progressive information retrieval from distributed image/video databases: The SPIRE project. In *Proc. of ECDL 98*, Crete, September 1998.

Captions

Figure 1

The level-1 progressive classifier under assumptions 1, 2 and 3. Figure (a) shows the decision regions Π_0^1, Π_1^1 and Π_{prog}^1. Figure (b) depicts the 4 decision regions of the progressive classifier, identified by the pair of produced labels. Figure (c) depicts the regions where the progressive classifier and the baseline (product) classifier produce different answers. For instance, in region α, the baseline classifier produces $\hat{\theta}_0 = 0$ and $\hat{\theta}_1 = 1$, while the progressive classifier produces $\hat{\theta}_0 = \hat{\theta}_1 = 1$. Figure (d) shows the decision regions of the best classifier that uses the pair $(X_1,\ X_2)$ as input.

Figure 2

The distribution of the coefficient W_1^4 and the components of the mixture under the Gaussian assumption, for the Markov model with $p = 0.08$. The dashed-dotted lines labeled A and B are the component densities of W_1^4 in the *homogeneous* regions, namely where all the labels θ_i are equal to 0 or 1, respectively. The dotted lines are the *inhomogeneous* component densities, when k of the labels are equal to 1, and the others are equal to zero, for $k = 1, 2, \ldots, 15$. The dashed line labeled C is the mixture density of W_1^4, when the coefficient corresponds to an inhomogeneous region.

Figure 3

The form of $J(\lambda) = R + \lambda t$ for the case of Theorem 1, plotted as a function of the threshold T and of λ.

Figure 4

A 310×240-pixel portion of the test-image. In figure (a) very dark regions are water bodies, dark regions are roughly agricultural and bright regions are roughly non-agricultural areas. Figure (b) shows the photointerpretation results, used as ground truth.

Comparison of Baseline classifier, ICM and Progressive Classifiers starting at level 1 and 2 in terms of speed and accuracy.

Classifier	classification time	classification accuracy	Speedup using 7-Nearest Neighbor	Speedup using CART
baseline	425.59	0.9844	1	1
ICM	430.21	0.9859	0.989	.293
Progressive, level 1	232.56	0.9863	1.83	1.44
Progressive, level 2	175.93	0.9842	2.42	2.98

Table 1: